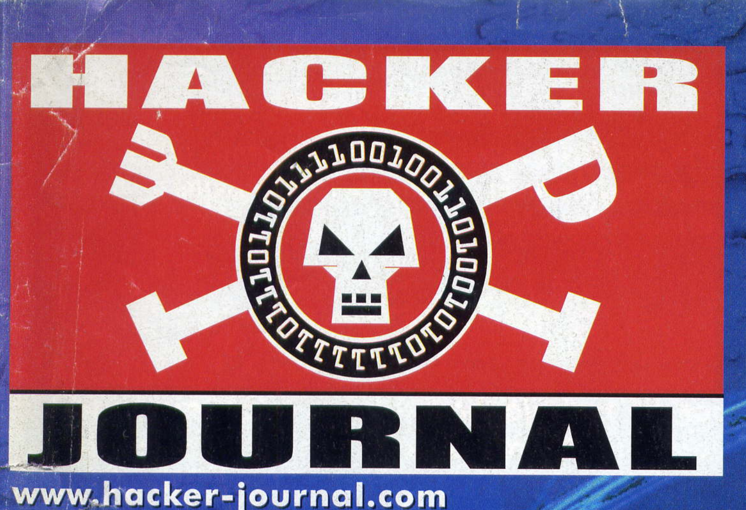


www.hacker-journal.com



2€
SIN PUBLICIDAD
SÓLO INFORMACIÓN Y ARTÍCULOS

EL PIRATA UCRANIANO

Nos relata las actividades radicadas en su país

SECRETOS DE SSL

Examinamos este protocolo de cifrado de amplia difusión

PRIMEROS PASOS CON LINUX

Guía de supervivencia para los que vienen del DOS

ZONAS DE DVD

Cómo enfrentarte a una restricción absurda

PROTOCOLOS DE INTERNET

Qué se cuece bajo las conexiones

SPOOFING DE ARP

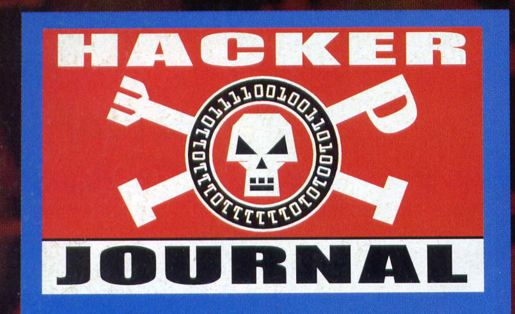
Cómo se simula una personalidad

3



4ever

>>> PRÁCTICA >>> SEGURIDAD >>> LINUX >>> LINKS



Año 1 - N. 3
Octubre 2003

Boss: theguilty@hacker-journal.com

Editor: grand@hacker-journal.com

Colaboradores: Bismark.it, Fabio Benedetti, Guglielmo Cancelli, Gaia, Nicola D'Agostino, Lele, Roberto "dec0der" Enea, >>>----Robin---->, Lidia,3d0, Mònica Batalla

Maquetación: Estudi Digital, S.L.

Diseño gráfico: Dopla Graphic S.r.l. info@dopla.com

Redacción
4ever S.r.l.
Via Torino, 51
20063 Cernusco S/N (MI)
Fax +39/02.92.43.22.35

Printed in Italy

Distribución
Coedis, S.L. - Avda. de Barcelona 225
08750 Molins de Rei (Barcelona)

Publicación bimensual registrada el 14/2/03 con el número MI2003C/001404

Director responsable:
Luca Sprea

hack·er (hăk'ər)

"Persona que se divierte explorando los detalles de los sistemas de programación expandiendo sus capacidades, a diferencia de muchos usuarios que prefieren aprender solamente lo mínimo necesario."

LA LUCHA POR LA VIDA

Obtener dinero por el trabajo honrado es algo que nadie cuestionaría seriamente. Otra cosa es cuando se quiere obtener una ganancia desmesurada y de por vida por algo que se hizo alguna vez. Y las cosas empeoran cuando, encima, se pretende que nadie más pueda hacer algo parecido al trabajo propio, para apartar así a la competencia y comerse el mercado hasta dejar mondos los huesos. Ya en la Edad Media se cualificó esta disposición de ánimo con un término muy claro: la usura.

Éste es el problema que se encuentra sobre la mesa en la actualidad más rabiosa de la informática. Nos estamos refiriendo a los bloqueos de mensajería instantánea que durante el mes de octubre han llevado a cabo Yahoo! y Microsoft. Utilizar excusas como la protección frente a la pornografía infantil para que la gente se rasque el bolsillo es algo tan insustancial que asusta a la inteligencia media. Curiosamente, en aquellos países en los que el servicio de mensajería instantánea ya es de pago, los males de la red desaparecen como por ensalmo y no es preciso dejar a nadie sin servicio.

Frente a tales desmanes, y para proteger la libertad colectiva, los defensores del código abierto siguen luchando a brazo partido para mantener abiertas las puertas de la comunicación en la que, de verdad, defienden como aldea global. Tal vez algún día, cada vez que intercambiemos información con alguien por la red, deberemos rendir un sentido tributo a estos desinteresados desarrolladores que trabajan por el bien de todos. Y si alguno de vosotros, lectores, tiene los conocimientos suficientes como para colaborar activamente, sería magnífico que echara un vistazo a los múltiples proyectos Open Source que ya están en marcha. ¡Este voluntariado merece la misma consideración que una ONG! La alternativa es muy sencilla: rascarse el bolsillo, pagar por todo y seguir engordando los mismos bolsillos de siempre. Ni siquiera IBM en sus mejores tiempos se atrevió a monopolizar tanto...

redaccion@hacker-journal.com

UNA REVISTA PARA TODOS


NEWBIE

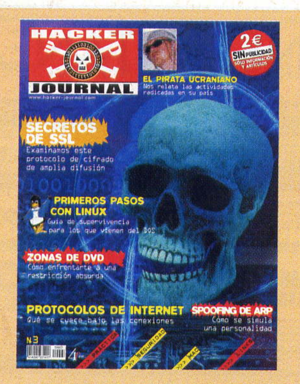

MID HACKING


HARD HACKING

El mundo hacker se compone de algunas cosas simples y otras complicadas. Hay curiosos, lectores sin experiencia y expertos para los cuales el ordenador no tiene secretos. Cada artículo de Hacker Journal está marcado con una clave para cada nivel: NEWBIE (para quien comienza), MIDHACKING (para quien ya está dentro) y HARDHACKING (para quien no existen los secretos).



Número 3 - Octubre 2003



¡SECRET ZONE!

He aquí los códigos para acceder a la Secret Zone de nuestro sitio, donde podréis encontrar información y utilidades interesantes. Con algunos navegadores, puede ser necesario insertar dos veces los mismos códigos. No os detengáis al primer intento

user: secret3
password: esdigital

EMPIEZA LA GUERRA DE LA MENSAJERÍA INSTANTÁNEA

LA PUGNA POR EL SOFTWARE ABIERTO

Los grandes proveedores de mensajería instantánea, Microsoft y Yahoo!, no quieren la compatibilidad cruzada entre soluciones. Su razonamiento se basa en el hecho que han invertido mucho dinero en su tecnología (cada uno en la suya, por supuesto) y para reunir bajo su ley a millones de usuarios felices, y opinan que ahora otros contendientes están intentando beneficiarse de su abnegado trabajo. No dudan en calificar de "competencia desleal" la actitud de quienes están dando servicio a los usuarios finales, permitiendo que estos puedan conectar con todos sus amigos y contactos, sea cual sea la solución utilizada.

Son muchos los fabricantes de software para mensajería instantánea que no piensan dar cancha a la pérdida de compatibilidad entre plataformas, emprendida por el bloqueo de MSN Messenger y de Yahoo! Mesenger. Entre estos fabricantes se puede citar a Cerullean, creador del software cliente de código propietario Trillian; los desarrolladores de Psi y Gaim, dos programas de mensajería de código abierto; Kopete, muy popular en los entornos Unix y Linux; y los creadores del protocolo Jabber, el protocolo de código abierto para mensajería instantánea (MI). También Apple trabaja con soluciones abiertas.

IBM aporta una solución intermedia. Su apuesta se basa en la tecnología SameTime, que en su día desarrolló Lotus. Se trata de un software que habilita la construcción de servicios de mensajería. Mediante SameTime cualquiera puede construir su propio servicio de mensajería instantánea. El servidor es el único producto de pago, mientras que los clientes pueden acceder con el software facilitado de forma gratuita, incluso desde dispositivos móviles. También puede utilizarse directamente un navegador Web. Otro punto a su favor es su integración con otras aplicaciones, como el correo electrónico, y actualmente los esfuerzos se orientan a integrar ambas cosas. Esta solución ha sido adquirida, entre otros, por Terra, con la que tiene montado su propio servicio.

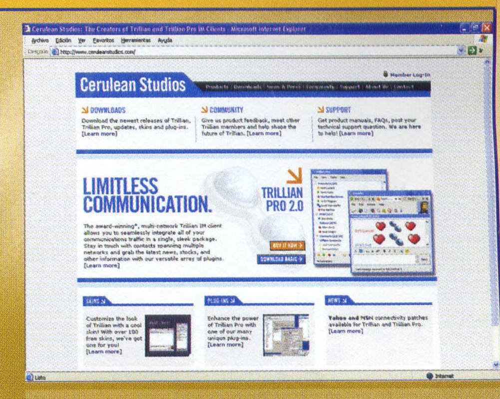
El mes de octubre ha vivido una guerra sin precedentes por la primacía en el campo de la mensajería instantánea. Las hostilidades fueron iniciadas por Microsoft, quien anunció el fin de la compatibilidad de sus servicios de mensajería instantánea con los de sus competidores, que hasta ahora se podían comunicar entre sí gracias a programas de otros desarrolladores. Este bloqueo debía iniciarse el 15 de octubre. Inmediatamente, Yahoo! reaccionó con energía, adelantando su propio bloqueo al día 1 de octubre

Estas operaciones no son en absoluto erráticas. Con el aislacionismo de la mensajería instantánea empieza en serio una carrera sin reglas para obtener rentabilidad de estos servicios. El secreto se encuentra en la falta de compatibilidad entre las distintas soluciones: no existe un protocolo normalizado para esta mensajería, a diferencia del HTML para la Web o el SMTP para el correo electrónico.

LA BATALLA DEL CÓDIGO ABIERTO

Mientras las soluciones propietarias medran en el río revuelto de los estándares, los partidarios del software Open Source se vienen esforzando en crear pasarelas que permitan compatibilizar los distintos programas de mensajería instantánea. Una vez más se repite la lucha entre las soluciones propietarias y los partidarios del software libre.

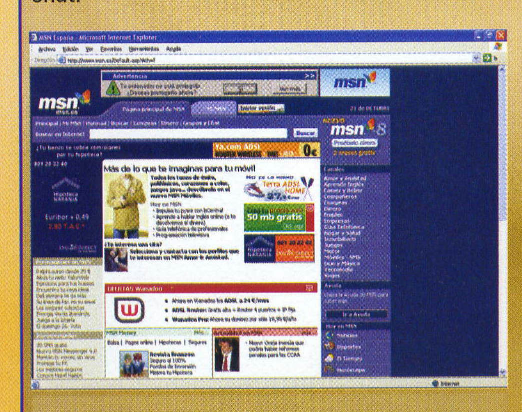
Como es sabido, la mensajería instantánea utiliza el texto como forma de diálogo en tiempo real entre usuarios conectados a Internet. Se trata de una tecnología que está experimentando actualmente una ascenso espectacular, como en su día vivieron otras soluciones ya consolidadas como la navegación Web o el correo electrónico. Sus puntos distintivos son la inmediatez, la facilidad de uso y la gratuidad. Precisamente la inmediatez lo convierte en el complemento ideal del correo electrónico, y su privacidad lo sitúa en perspectiva respecto al chat.



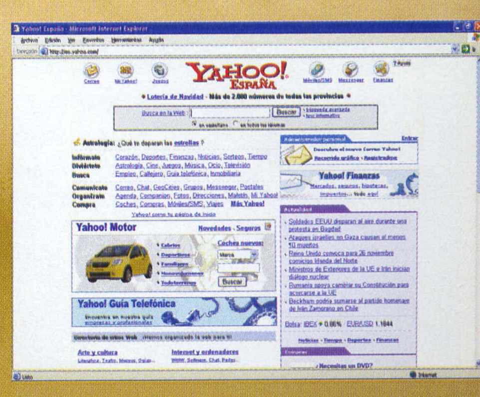
Trillian significa una solución a la incompatibilidad de caracteres.

Algunos analistas afirman que la mensajería instantánea está madura para su eclosión en aspectos como la seguridad, mientras que mantiene abiertos interrogantes como la rentabilidad y la compatibilidad. La rentabilidad, que en principio debería dejar aparte al usuario doméstico –al menos es la aspiración de los partidarios del código abierto–, debería reflejarse al menos en los entornos profesionales, para dar soporte a un uso licenciado con soporte técnico. Sin embargo, el gran caballo de batalla es la compatibilidad. A diferencia del correo electrónico, que nació con el estándar SMTP fuertemente implantado, la mensajería instantánea se ha visto sometida desde su inicio a la separación entre los distintos proveedores del servicio: cada uno ha aspirado a aumentar su propia base de usuarios y a asegurarse de que "el que se mueva no sale en la foto". Mientras los servicios han sido gratuitos, la solución más utilizada ha sido la instalación de los diversos programas disponibles para conectar con el máximo de gente. Sin embargo, si ahora empieza a difundirse la costumbre de cobrar por el uso, esta opción dejará de ser viable: pocos estarán dispuestos a pagar varias veces por el mismo servicio, simplemente porque sus propietarios están intentando obtener réditos de su base instalada.

La incompatibilidad es el gran freno a la de-



Microsoft se siente fuerte con su solución integrada MSN...



...pero Yahoo! no ha querido quedarse atrás en la pugna.

finitiva difusión de la mensajería instantánea. A pesar de ello, pocas veces se habla de establecer estándares en este campo. Y hasta ahora estos contactos han fracasado, no por razones técnicas sino comerciales. Es de esperar que con la guerra iniciada por Microsoft y otros será el momento de la intervención decisiva del IEFT, el gran responsable de la estandarización en Internet.

QUIÉN SE COMERÁ EL PASTEL

Como siempre, Microsoft cuenta con la baza de la implantación de Windows y otros servicios complementarios, como Hot Mail y MSN. Sin duda cuenta con una ventaja importante, pero sus competidores no han tirado la toalla. AOL (que posee ICQ), Yahoo! Messenger, Apple o Terra, así como muchos pequeños proveedores, se mantienen en la pugna, al menos mientras no sean absorbidos por los peces más grandes de este nuevo mar...



El histórico ICQ pertenece desde hace unos años a AOL.

Apple apuesta de nuevo con el iChat por el diseño cuidado.

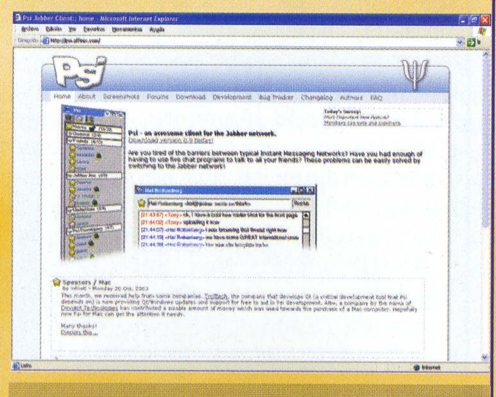
Muchos desarrolladores independientes han desarrollado soluciones basadas en pasarelas que les han permitido encontrar su nicho en la mensajería instantánea. saltan estas barreras. Uno de los principales es Trillian, un programa gratuito de Cerulean que funciona bajo Windows y MacOS. Trillian tiene su valor en una única interfaz desde la que se pueden intercambiar mensajes entre miembros de AOL, MSN y Yahoo! Con Trillian, un usuario podía conectar con otro usuario de MSN Messenger de Microsoft , y con un tercero suscrito a AOL.

El mes de octubre ha intentado certificar la defunción de estas soluciones. Obviamente, los desarrolladores de plataformas cruzadas se han apresurado a anunciar (y a trabajar en) nuevas compatibilidades contra las nuevas fronteras. Atención, usuarios: se acercan tiempos de cambios constantes de versión en todos los programas para, por una parte, erigir barreras y, por otra, saltarlas. La primera constatación es que en el camino se perderán montones de contactos, y que deberán ser los usuarios quienes, con paciencia, se dediquen a rehacer los lazos entre sí. Se dará una situación que dotará de nuevas fuerzas a los defensores del Open Source, quienes lucharán una vez más por la liberalización de las nuevas tecnologías. Una situación repetida mil veces, ahora en el campo de la mensajería instantánea. Si el éxito de soluciones como Linux se repite en este campo, se avecinan emociones fuertes.

BARRERAS LEGALES

Las soluciones Open Source tienen que utilizar la ingeniería inversa para mantener la compatibilidad cruzada, pues los barreneros de la comunicación abierta, Microsoft y Yahoo!, tienen sus nuevos protocolos encerrados bajo siete llaves. A pesar de tantas precauciones, la mayoría de desarrolladores de código abierto anuncian que ya están preparados para afrontar los cambios. Una vez más, la posesión de código secreto parece haber servido de bien poco.

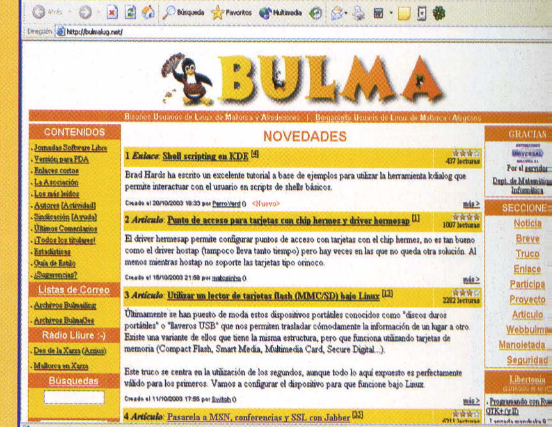
Pero si hay alguien experto en trasladar las pugnas al campo legal es Microsoft, quien añadirá a la novedad del protocolo una licencia de uso con más cláusulas restrictivas. Con la nueva licencia, el usuario se comprometerá a usar sólo el software de mensajería de MSN Messenger o Windows Messenger, o bien una aplicación homologada por Microsoft. A fecha de hoy, sin embargo, no existe aún ninguna solución de terceros admitida en el nuevo corralito de la mensajería instantánea, amenazada con mantenerse en su incompatibilidad congénita.



PSI es uno de los posibles clientes compatibles con Jabber.

⊃ SOLUCIONES ABIERTAS

Con el uso de las pasarelas, puedes abrir tu MSN Messenger para comunicar con una amiga que utiliza AOL. Con el uso intermedio de software como Trillian, la comunicación era transparente. Sin embargo, con los bloqueos emprendidos en octubre por Microsoft y Yahoo!, este paraíso empieza a perder habitantes.



En Mallorca existe un activo sitio Web, mantenido por un grupo de usuarios de Unix. Su nombre es Bulma, y en su interior se ha instalado un servidor que funciona sobre el protocolo de código abierto Jabber. Los usuarios descargan el software cliente, como SPI, GAIM, Kopete o Trillian, y pueden conversar entre ellos y con los de AOL, Yahoo! y MSN Messenger gracias a las correspondientes pasarelas que implementa Jabber. Los usuarios se descargan el software PSI, por ejemplo. El programa pide que se registre el nombre o apodo en un servidor un servidor Jabber, en esta caso el de Bulma (bulmalug.net) y le asigna una dirección universal de Jabber, por ejemplo, tunombre@bulmalug.net. Una opción del programa PSI muestra las pasarelas a distintos servicios. Para registrarte en la pasarela tienes que darte de alta antes en la misma dirección de Messenger, AOL o Yahoo.

Con Jaber cualquiera puede instalar su propio servicio de mensajería instantánea. Además de ser gratis, es un sistema distribuido, siguiendo el modelo de otros programas peer-to-peer como eDonkey o Kazaa –salvando las distancias por sus distintas finalidades–, de modo que no necesita grandes instalaciones como los grandes servidores centralizados de mensajería instantánea. Al ser una solución abierta, no se encuentra sometida a las veleidades monopolísticas de ningún fabricante en particular.

HOT!

PRESCOTT DEBUTARÁ EL 3 DE DICIEMBRE

El 3 de diciembre se introducirá la próxima generación de procesadores Intel, con el nombre en clave Prescott. La nueva CPU, que incluye varias innovaciones respecto al proyecto Pentium 4, será la primera para sistemas de sobremesa basada en el proceso productivo de 0.09 micras. Prescott integrará 1 MB de caché L2 y se introducirá inicialmente en versiones a 3,4 y 3,2 GHz de velocidad de reloj, con una versión a 3,6 GHz prevista para el primer trimestre de 2004. En el segundo trimestre del año 2004 Intel introducirá una nueva versión del procesador Prescott, basada en el Socket LGA 775; junto con el nuevo Socket Intel introducirá también una nueva versión del procesador Prescott con una frecuencia de reloj de 3,8 GHz. Para el mercado entry level Intel introducirá, a partir del segundo trimestre de 2004, una versión de la CPU Celeron basada en un core a 0.09 micras de proceso productivo, con una memoria caché L2 de 256 KB.

CARRETERAS INTELIGENTES EN CANADÁ

El gobierno canadiense ha mantenido una serie de conversaciones con las administraciones locales para verificar la posibilidad de instalar en las principales arterias de la red viaria un sistema nacional de sensores para la información meteorológica. La idea en la base del proyecto es la de crear carreteras "inteligentes" en las que los automovilistas podrán saber en cualquier momento las condiciones del tiempo y las del firme. Se dedicará una particular atención a la información respecto al peligro de heladas. Los sensores serán también útiles para las empresas de mantenimiento de las carreteras, con la posibilidad de establecer sobre la marcha, por ejemplo, la necesidad de echar sal para evitar los problemas de nieve.

NUEVOS LECTORES DE MP3 PARA EL OTOÑO

El mercado de los lectores de MP3 portátiles está viviendo una expansión muy elevada. Lo demuestran los nuevos productos presentados recientemente en el mercado. En primer lugar se sitúa Creative, que para el otoño propone tres nuevos modelos, con dimensiones y capacidad de archivo muy distintas entre sí. El primer producto de la nueva gama es el MuVo NX, con capacidad de archivo de 128 MB y el añadido de una pantalla LCD monocroma donde se muestra el título de la pieza, su duración y otras funcionalidades. Un pequeño micrófono integrado permite grabar hasta un máximo de ocho horas de discurso de voz. MuVo2 (Cube) es la nueva propuesta de gama media: con unas dimensiones muy reducidas, permite almacenar hasta 1,5 GB de datos mediante un microdrive integrado. La interfaz es de tipo USB 2.0 y permite utilizarlo también como sistema de memoria para archivos temporales y transferencia de datos. El modelo de gama más alta es el Jukebox Zen NX, dotado de 30 GB de capacidad de archivo, de dimensiones y peso más reducidos respecto al modelo original. La batería permite utilizar el Zen NX hasta un máximo de catorce horas en playback. Las interfaces de conexión son USB 2.0 y Firewire. También presenta novedades Anubis, que ha lanzado el nuevo Typhoon Live Music Mp3 Player w/Fm, un dispositivo que reúne en su interior cuatro funcionalidades: lector de Mp3, grabadora, radio FM y disco duro externo para transportar tus propios datos en cualquier situación y conectarlo a un PC o portátil mediante la conexión USB. Existen tres versiones disponibles, dotadas respectivamente con 64, 128 y 256 MB de memoria. Además de las características anteriores, la función de grabación en directo de 4 a 16 horas, la pantalla LCD retroiluminada con una luz azul en la que se puede ver el estado de las funciones o bien un ecualizador de cinco bandas. El dispositivo va equipado con un software para el reconocimiento de textos (que funciona a través del pc), el cual lee textos y los reproduce en modalidad de audio a través del reproductor. Sus dimensiones son reducidas: unos 30 gramos de peso. Para mayor información: www.anubisline.com. Finalmente, y no por ser el último es menos importante, Apple ha actualizado su línea propietaria iPod. Los modelos anteriores de 10, 15 y 30 GB han sido sustituidos por otros de 20 y 40 GB. El modelo de 40 GB permite almacenar hasta 10.000 títulos con una calidad próxima al CD, todo ello en un dispositivo extremadamente portátil y compacto, con un pseo de tan sólo 176 gramos. iPod cuenta con la funcionalidad Auto-Sync, que descarga la biblioteca musical del Mac o del PC y la actualiza de modo totalmente automático cada vez que se conecta el dispositivo al sistema. Los nuevos iPod para Mac y Windows están disponibles en los distribuidores autorizados o en la Apple Store (www.apple.com/spainstore). Más información en el sitio web de Apple www.apple.es.

EL GRAN HERMANO

Llega el gran Hermano! El largo brazo del control de mamá y papá se apresta a convertirse en un monstruo tentacular. Todo por culpa de la tecnología.

Los teléfonos móviles nos hacen localizables en todo momento, las páginas electrónicas y el registro de las operaciones en línea no nos permiten ya corrernos según que juertas. Y ahora, desde los Estados Unidos, nos está llegando la RS-1000 Black Box, una especie de caja negra que se instala en el coche y que registra todas las jugarretas automovilísticas, por fortuna sólo éstas, de los hijos descarriados. Veremos cómo se memorizan las infracciones de los límites de la velocidad, de los cinturones de seguridad incorrectamente sujetados, las curvas tomadas con excesiva brusquedad.

Esto significa que cada vez que en lugar de ir a la biblioteca a estudiar, nos vayamos de paseo al parque o salgamos con la novia, mamá llegará a saberlo. Una auténtica plaga. Pero esto no es todo. Para devolver al redil a los conductores demasiado deportivos, la Black Box dispone de una señalización acústica que advierte la infracción. De nada sirve subir el volumen de la radio: el volumen del gendarme electrónico subirá en consecuencia. Es el fin. A menos que se imponga el viejo principio: "a grandes males, grandes remedios": ¿cuánto tiempo tendrá que pasar hasta que la caja negra sea hackeada? Se aceptan apuestas.

EL SHUTTLE VOLVERÁ A VOLAR



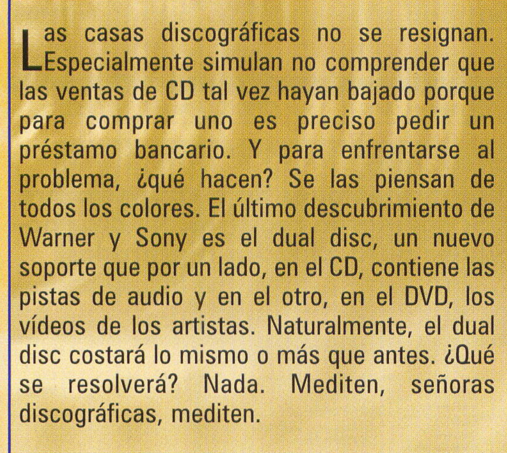
Será marzo el mes en el que el Shuttle volverá a volar. Lo ha anunciado la NASA, la oficina espacial estadounidense. El lanzamiento debería darse entre el 11 de marzo y el 6 de abril, pero si las normas de seguridad no se cumplen, se aplazará por tiempo indefinido. La NASA no quiere caer de nuevo en la acusación recibida recientemente, el febrero pasado, cuando siete astronautas perdieron la vida, según parece a causa de un agujero en el ala del Shuttle causado por un panel de goma aislante que se desprendió de la nave tras el despegue. En esa ocasión, los miembros del Columbia Accident Investigation Board (Caib) castigaron duramente la política de la NASA, acusándola de ignorar la seguridad de los propios miembros en favor de la continuidad de sus vuelos espaciales. La NASA ha rediseñado la nave espacial para hacerla más segura y ha asegurado que aplicará cambios sustanciales al Atlantis, para eliminar estos problemas. Actualmente se están probando materiales y procedimientos nuevos.

NUEVOS CDS, LA MISMA ESTAFA

Las casas discográficas no se resignan. Especialmente simulan no comprender que las ventas de CD tal vez hayan bajado porque para comprar uno es preciso pedir un préstamo bancario. Y para enfrentarse al problema, ¿qué hacen? Se las piensan de todos los colores. El último descubrimiento de Warner y Sony es el dual disc, un nuevo soporte que por un lado, en el CD, contiene las pistas de audio y en el otro, en el DVD, los vídeos de los artistas. Naturalmente, el dual disc costará lo mismo o más que antes. ¿Qué se resolverá? Nada. Mediten, señoras discográficas, mediten.



ORDENADORES MÁS POTENTES GRACIAS AL ADN



Un ordenador que utiliza la estructura del ADN. Esta idea se encuentra en la base del presupuesto de que las celdas humanas y los procesadores de los ordenadores conservan la información y efectúan procesos sobre ella de manera muy similar. La estructura que se deriva de esta idea resultaría ser más potente que las de los procesadores utilizadas hasta ahora. El razonamiento que se encuentra en la base del proceso es simple: los ordenadores utilizan una cadena compuesta por números 1 y 0 para guardar la información, mientras que los seres vivos utilizan moléculas representadas por las letras A, T, C y G. Los científicos Milan Stojanovic y Darko Stefanovic han realizado, al hilo de esta idea, Maya, una máquina insuperable en el juego del Tris que utiliza una compleja mezcla de enzimas de ADN para tomar sus propias decisiones. Según los investigadores, si se organizan de manera diferente y más compleja las enzimas, la potencialidad es infinita. El valor de este dispositivo es el hecho de ser una estructura interactiva que no requiere de la intervención humana para tomar sus propias decisiones. Desarrollar un sistema que utilice las moléculas del ADN permitiría hacer los ordenadores aún más pequeños.

HOT!

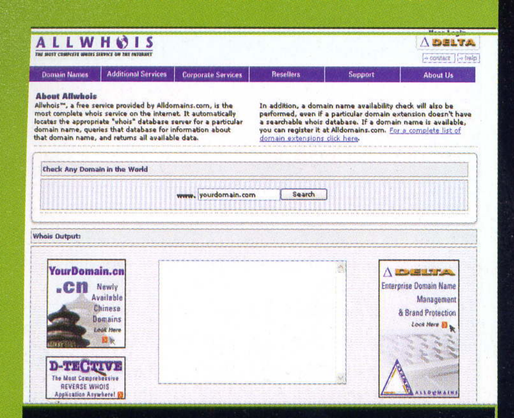
EPIPHANY, LA ALTERNATIVA A MOZILLA

Se llama Epiphany y es la alternativa más "soft" a Mozilla. Se trata de un browser open source que, tras varios meses de desarrollo, ha llegado a su primera versión estable, la 1.0. Fruto de un proyecto fundado por el italiano Marco Pesenti Gritti, ex de Galeon, otro browser libre para Linux cuyo código ha servido de base para el nacimiento de Epiphany. El objetivo es ofrecer a los usuarios de Linux un browser que, aún basado en el núcleo de Mozilla, renuncia a las funciones que hacen farragoso a éste. El motor de rendering es Gecko y la interfaz aprovecha la integración con el tema del escritorio Gnome (del cual se ha convertido en una de las aplicaciones oficiales), una característica que, si bien limita fuertemente la posibilidad de portabilidad a otros sistemas operativos, proporciona al browser coerencia gráfica y funcional con las demás aplicaciones de Gnome.

¡PERO QUÉ FANTASMAS... SON LOS ULTRASONIDOS!

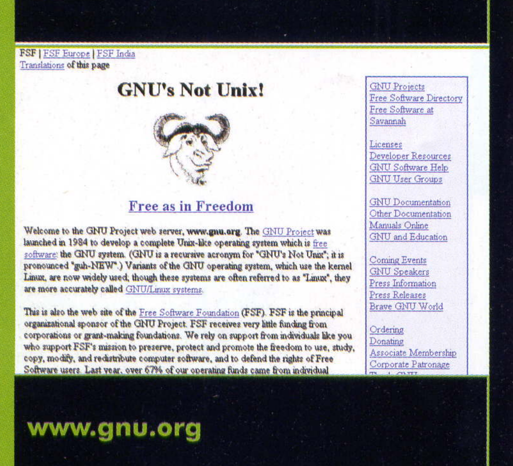
Un experimento llevado a cabo por algunos científicos británicos ha demostrado que en presencia de ultrasonidos, sonidos no percibidos por el oído humano, el organismo humano reacciona de modo raro. El doctor Richard Lord y el profesor Richard Wiseman han comprobado el impacto de los ultrasonidos en un concierto con un público de 750 personas. Los dos científicos han dado a escuchar a su audiencia cuatro piezas, dos de las cuales contenían ultrasonidos. Sin saber cuáles de las piezas habían sido "retocadas", el 22% de los entrevistados declaró que al escuchar las piezas con ultrasonidos había experimentado sensaciones desagradables, melancolía, escalofríos, y hasta emociones más fuertes, como rabia y miedo. Según algunos científicos, este tipo de sonidos podría hallarse en lugares considerados infestados de fantasmas. Los ultrasonidos se producen de forma espontánea en la propia naturaleza, como en presencia de una tempestad, con vientos muy violentos y con algunos tipos de terremotos. Diversas razas de animales, entre ellas los elefantes, se sirven de ellos para comunicarse a través de largas distancias.

Los clásicos internacionales



www.allwhois.com

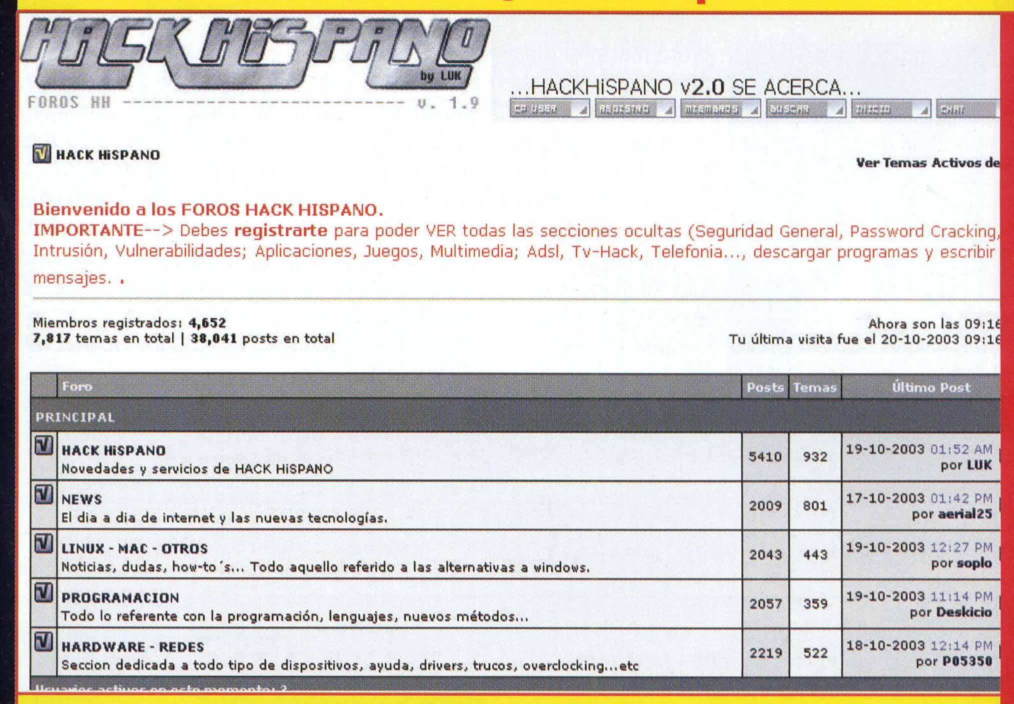
¿Qué sería de la gran telaraña mundial sin un buen Quién es Quién? En este sitio puedes preguntar por cualquier nombre de dominio para conocer quién es su propietario y todos los datos disponibles asociados. Es capaz de buscar páginas web de cualquier lugar del mundo y con cualquier tipo de extensión (.com, .es, .net, etcétera).



www.gnu.org

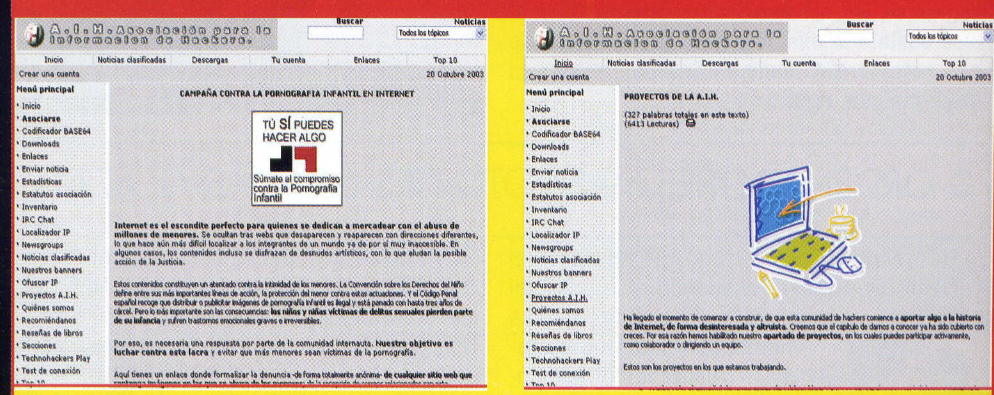
Un gran clásico donde los haya, la página de GNU está dedicada a los proyectos basados en la licencia del mismo nombre. Como bien sabrás, se trata de proyectos que se basan en la colaboración libre para producir programas de libre distribución. Esta valerosa alternativa a los fabricantes que sólo buscan enriquecerse a nuestra costa merece todos nuestros elogios, y en su interior se reúnen los mejores programadores que pueblan la tierra. Obligado.

¡15 minutos de gloria! ¿Queréis que vuestr ¡Enviadlo a redaccion@hacker-journal.com!



www.hackhispano.com

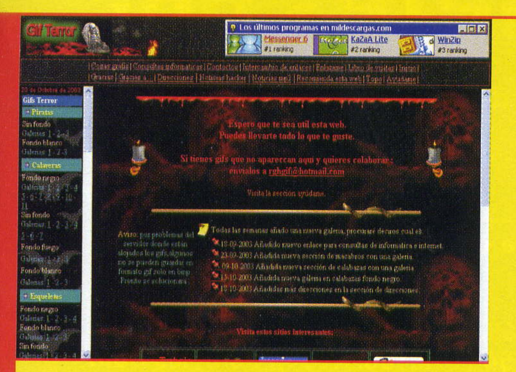
Un sitio repleto de foros donde encontrar montones de mensajes sobre los temas más diversos del hacking. Si además te registras, tendrás acceso a una serie de secciones reservadas para usuarios registrados



www.infohackers.org

¡Sitios web como éste dan prestigio a nuestra actividad! Aquí encontrarás información de calidad y actualizada, claramente estructurada en unas páginas de magnífico diseño. Y si eres de los que se sienten con ganas de colaborar, no te faltarán propuestas y proyectos en los que aportar tu granito de arena. Y si lo que pretendes es mantenerte al día, además de leer esta revista, encontrarás noticias clasificadas por categorías. Se impone una visita.

sitio web aparezca entre los reseñados?





www.gifterror.com

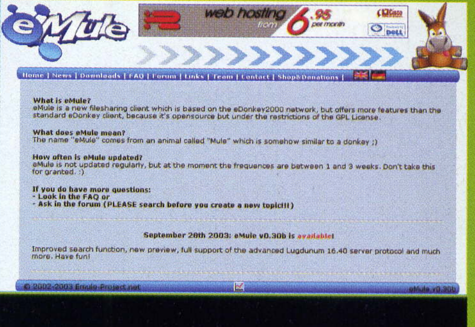
Hay días en los que la creatividad no se muestra a la altura, ¿verdad? Esto no es un gran problema si sabes dónde acudir para recabar ayuda. Este sitio es especialmente adecuado para cuando necesites elementos gráficos con un punto hacker o gore y no sepas qué poner. Simplemente pasea por ahí y toma lo que necesites. Y si tienes alguna creación propia, haz tu propia aportación.



www.flashkit.com

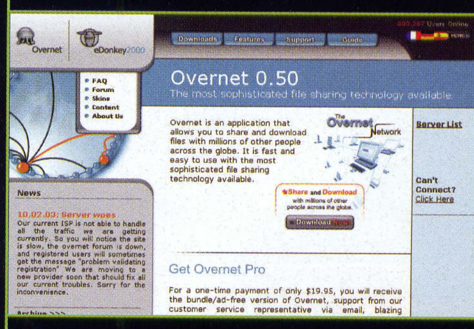
Las páginas web no son lo mismo desde que el formato gráfico Flash de Macromedia ha invadido las páginas. Desde geniales animaciones hasta potentes gráficos interactivos, todo está al alcance del webmaster con ganas de crear un sitio interesante. Aquí encontrarás componentes para Flash, así como las últimas noticias sobre este magnífico entorno de animación. Y es posible encontrar también recursos terminados y listos para su uso.

Los clásicos internacionales



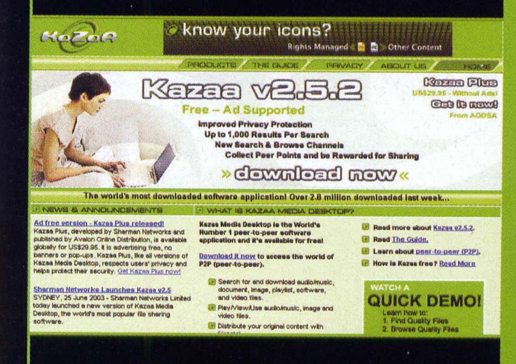
www.emule-project.net

Los programas de intercambio de archivos cuentan con una enorme popularidad. Ésta es la página oficial del clónico del edonkey.



www.edonkey2000.com

Y si existe un clónico, sin duda hay que visitar al original. Éste es el sitio web oficial del edonkey, para quienes rechazan imitaciones. También es necesario si te apuntas a Overnet, ya que ambos proyectos se llevan desde el mismo sitio.



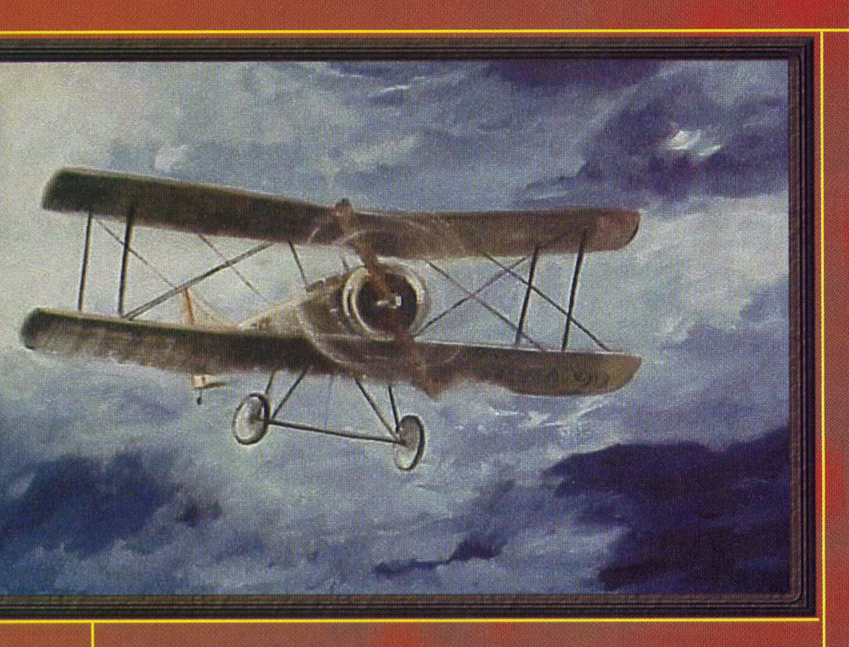
www.kazaa.com

Y si hablamos de edonkey y emule, es necesario hablar también de Kazaa, probablemente el número uno en cantidad de usuarios.

CÓMO DESVIAR PAQUETES DENTRO DE UNA RED LOCAL

Spoofing de los paquetes ARP

Vamos a explicar cómo es posible realizar ataques directamente sobre el mecanismo de distribución de paquetes en una red local.



Cuál es la mayor vulnerabilidad de una red local? Probablemente, la posibilidad de falsificar paquetes ARP. Por este medio, se puede hacer creer que ciertos equipos pertenecen a una red dada, sin ser ello cierto, redireccionando TODO el tráfico de Ethernet. ¿Cómo puede explotarse esta vulnerabilidad? En muchos casos, un atacante la utilizará para monitorizar el tráfico de la red. O podría utilizarla para un ataque del tipo Denial of Service, o para interponerse en una comunicación, interceptándola (ataque del tipo "man in the middle").

>> ¿QUÉ ES EL ARP?

ARP significa Address Resolution Protocol, y se usa para mapear las direcciones IP en direcciones de Ethernet (MAC). Cuando se transmite un paquete IP a una red, el sistema tiene que saber a qué equipo conectado físicamente a la red local debe enviarlo (si al enrutador o a otro equipo de la red). Por consiguiente, "pregunta" a la red quién tiene el IP x.x.x.x y alguien responderá x.x.x.x si encuentra el puesto de red que cuenta con la dirección MAC xx.xx.xx.xx.xx.xx. De este modo se podrá completar el header datalink (802.3) y podrá enviarse el paquete. Este método es parecido al DNS, que sirve para asociar el número IP a una cierta dirección del tipo nombreequipo.nombredominio.es. Si quiero enviar un paquete a nasa.gov, "pregunto" (en este caso a la autoridad NS de nasa.gov) el ip que corresponde a nasa.gov. Así puedo completar la cabecera IP y enviar el paquete. Hay dos tipos de paquetes arp: arp request y arp reply. Ilustremos el concepto con tcpdump:

```
192.168.1.2 quiere enviar un icmp echo a
192.168.1.154:

# ping -c 1 192.168.1.154
PING 192.168.1.154 (192.168.1.154): 56 bytes data
64 bytes from 192.168.1.154: icmp_seq=0 ttl=255
time=3.0 ms

tcpdump:
15:19:26.217004 0:10:a4:c0:15:92 ff:ff:ff:ff:ff:ff 0806 42:
arp who-has 192.168.1.154 tell 192.168.1.2
15:19:26.217563 0:80:c8:7a:39:14 0:10:a4:c0:15:92
0806 64: arp reply 192.168.1.154 is-at 0:80:c8:7a:39:14
15:19:26.217608 0:10:a4:c0:15:92 0:80:c8:7a:39:14
0800 98: 192.168.1.2 > 192.168.1.154: icmp: echo
request (DF)
15:19:26.218351 0:80:c8:7a:39:14 0:10:a4:c0:15:92
0800 102: 192.168.1.154 > 192.168.1.2: icmp: echo
reply
```

El primer paquete es: "decir a 192.168.1.2 el mac de 192.168.1.154". Obviamente, es un paquete broadcast (ff.ff.ff.ff.ff.ff) porque está "buscando" el host.
El host responde con un paquete unicast que dice "192.168.1.154 se encuentra en la dirección 0:80:c8:7a:39:14".
En este momento se puede mandar el paquete ICMP, igual que sucede con DNS cuando alguien ejecuta telnet nasa.gov. Primero se envía el paquete UDP al puerto 53 y luego el syn al 23. Si ahora se envía otro ICMP a 192.168.1.154, observa que NO se realizará una nueva consulta ARP. Los ARP, como los hosts de los NS, se guardarán en memoria caché:

```
root:~# arp -na
 (192.168.1.154) at 00:80:C8:7A:39:14 [ether] on eth0
root:~#
```

Todos estos datos en la caché "caducan", y por ello es preciso enviar una nueva consulta. Naturalmente, la caché sirve para no sobrecargar la red con paquetes ARP.

>> Cómo se forman los paquetes ARP

Veamos cómo se construye un paquete ARP; el código está tomado de /usr/include/linux/if_arp.h:

```
struct arphdr
{
unsigned short  ar_hrd;  /* format of hardware address */
unsigned short  ar_pro;   /* format of protocol address */
unsigned char  ar_hln;  /* length of hardware address */
```





```
unsigned char  ar_pln; /* length of protocol address  */
unsigned short  ar_op; /* ARP opcode (command)  */

#if 0
        /*
*Ethernet looks like this : This bit is variable sized
however...
*/
unsigned char  ar_sha[ETH_ALEN]; /* sender hardware
address     */
unsigned char  ar_sip[4]; /* sender IP address
*/
unsigned char  ar_tha[ETH_ALEN]; /* target hardware
address     */unsigned char  ar_tip[4]; /* target IP address
*/
#endif

};
```

>> ARP Reply

Format y length no nos interesan (arp NO sólo sirve para asociar direcciones ethernet e IP: es un protocolo genérico, pero aquí sólo nos centraremos en IP). Ar_op es ARP request o reply. "Sender hardware address e IP" (dirección MAC de quien envía la consulta e IP correspondiente), y "target hardware e ip" son las partes más interesantes del paquete. Sender hardware y Sender IP son el IP y el MAC de quien envía el paquete. Sin embargo, si el paquete es request, target hardware se rellena con ceros (porque no se conoce) y target ip contiene el ip de quien queremos saber la dirección de hardware. En la respuesta simplemente llega rellenado el target hardware, con el opcode modificado.

El sistema operativo podrá enviar los paquetes mediante el guardado en la tabla ARP (en caché). Esta tabla se actualiza y cambia mediante los paquetes ARP. Empezamos a enviar algunos paquetes fingidos y veamos qué sucede... En teoría, cuando se envia una ARP request, si mando un reply con MI dirección mac, el sistema se cree que se conecta a X pero en realidad se conecta conmigo. En pocas palabras, si yo envío un NS reply que afirma que la dirección de nasa.gov es 192.168.1.2, el host se cree que se conecta a nasa.gov pero en realidad se conecta a 192.168.1.2).

Veamos un ejemplo sencillo:

192.168.1.154 (la víctima) se quiere conectar a 192.168.1.2 (que en realidad no existe).

```
root@DigitalF:~# ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 bytes data
```

no devuelve nada porque no recibe ARP reply

```
29:36:41.323528 0:80:c8:7a:39:14 ff:ff:ff:ff:ff:ff 0806 64:
arp who-has 192.168.1.2 tell 192.168.1.154 (sin reply)
```

en efecto:

```
root@DigitalF:~# arp -na
 (192.168.1.2) at <incomplete> on eth0
 (192.168.1.1) at 00:10:A4:C0:15:92 [ether] on eth0
   root@DigitalF:~#
```

en su lugar intentamos enviar un reply falso de 192.168.1.1 (que tiene un mac 00:10:A4:C0:15:92):

```
 # ./arp <dev> <srcmac> <dstmac> <arp op:1req
 2 rep> <srcmac> <srcip> <dstmac> <dstip> <delay>
```

obteniendo

```
# ./arp eth0 aa:aa:aa:aa:aa:aa ff:ff:ff:ff:ff:ff 2
00:10:A4:C0:15:92 192.168.1.2 aa:bb:bb:bb:bb:bb
192.168.1.30 10000000
DELAY = 10000000
SENT
#

root@DigitalF:~# arp -na
? (192.168.1.2) at 00:10:A4:C0:15:92 [ether] on eth0
? (192.168.1.1) at 00:10:A4:C0:15:92 [ether] on eth0
root@DigitalF:~#
```

Ahora 192.168.1.154 (DigitalF para entendernos...) cree que 192.168.1.2 es 00:10:A4:C0:15:92

ahora intentamos transmitir...

```
root@DigitalF:~# ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 bytes data
```

```
--- 192.168.1.2 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
root@DigitalF:~#

tcpdump:
20:08:02.816143 0:80:c8:7a:39:14 0:10:a4:c0:15:92
0800 102: 192.168.1.154 > 192.168.1.2: icmp: echo request
```

>> ARP Request

Como queríamos demostrar: en pocas palabras, hemos suplantado a 192.168.1.2 (que al principio no existía) y no hay respuesta porque el sistema operativo sólo conoce a .1 y no a .2. Por consiguiente, el MAC es nuestro y el dst IP se mantiene intacto. Así que cuidado, si observas valores extraños en la línea de comandos... source mac aa:aa:aa:aa:aa:aa no encaja con 00:10:A4:C0:15:92; esto ocurre porque el kernel no lleva a cabo una verificación. Éste es sólo uno de los ejemplos posibles, pero un atacante podría hacer muchas más cosas falsificando los reply. Por citar sólo algunas, digamos que con las ARP reply es posible modificar la caché ARP, y enviar paquetes broadcast sin que el target ip sea controlado.
¿Para qué sirve enviar un ARP request?

```
# ./arp eth0 aa:aa:aa:aa:aa:aa ff:ff:ff:ff:ff:ff 1
00:10:40:30:20:11 192.168.1.2 00:00:00:00:00:00
192.168.1.8 10000000
DELAY = 10000000
SENT
#
```

ahora veamos la caché de DigitalF...

```
root@DigitalF:~# arp -na
 (192.168.1.2) at 00:10:40:30:20:11 [ether] on eth0
 (192.168.1.1) at 00:10:A4:C0:15:92 [ether] on eth0
root@DigitalF:~#
```

Nuevamente, hemos cambiado la caché por 192.168.1.2. Pero ¿por qué? Sigamos el protocolo... Los arp request contienen el source ip y source mac de un host y el dst ip a quien se ha enviado la consulta. ¿Por qué no guardar en caché el source y dest mac de las consultas que recibamos? Esto permitirá evitar el envío de un request para ese IP en el futuro. De hecho, este procedimiento forma parte del protocolo. Un atacante podría insertar datos falsos para guardar en caché lo que mejor le parezca. Se observa que también en este caso el dst ip no se controla, y el paquete es broadcast.

>> Posibles ataques

Intentamos enviar un request con un source ip que ya no se encuentra en la tabla (es decir, que realmente intentamos crear una entrada en la tabla arp).

```
# ./arp eth0 aa:aa:aa:aa:aa:aa ff:ff:ff:ff:ff:ff 1
00:10:40:30:20:11 192.168.1.4 00:00:00:00:00:00
192.168.1.8 10000000
DELAY = 10000000
SENT
#
root@DigitalF:~# arp -na
 (192.168.1.1) at 00:10:A4:C0:15:92 [ether] on eth0
root@DigitalF:~#
```

No ocurre nada. Pero si se intenta con el verdadero dst ip de DigitalF (.154)

```
# ./arp eth0 aa:aa:aa:aa:aa:aa ff:ff:ff:ff:ff:ff 1
00:10:40:30:20:11 192.168.1.4 00:00:00:00:00:00
192.168.1.154 10000000
DELAY = 10000000
SENT
#
root@DigitalF:~# arp -na
 (192.168.1.4) at 00:10:40:30:20:11 [ether] on eth0
 (192.168.1.1) at 00:10:A4:C0:15:92 [ether] on eth0
root@DigitalF:~#
```

¡Y voilà! Se ha creado la entrada. Se observa que con arp reply no funciona si se inserta el verdadero dst ip. Con dst ip 127.0.0.1, 214.0.0.1 etcétera, no funciona. Por tanto, el kernel realmente controla el dst ip para crear una entrada en la tabla, y no se puede difundir por broadcast. Por el contrario, se pueden difundir perfectamente paquetes que actualicen la caché y crear entradas en ella (la tabla debe contener el dst ip correcto, por lo que no podemos difundir cuando creamos entradas).
Por broadcast se observa que con un paquete un atacante puede pasar inadvertido. Modificando la tabla de ARP un atacante podría redirigir todo el tráfico de la red a su propio equipo, capturarlo, y luego enviarlo eventualmente al verdadero destino. Supongamos que tenemos dos equipos en una red local, A y B, que tienen en caché la dirección MAC del enrutador. Si se modifica, insertando fraudulentamente la dirección MAC del equipo C en su lugar, todo el tráfico destinado al enrutador (en la práctica, todo el tráfico de Internet) será redirigido a C. Éste, a su vez, podrá redirigirlo al enrutador o hacer él mismo de enrutador, pero con la posibilidad de interceptar las comunicaciones y alterar cualquier paquete.

Para saber más

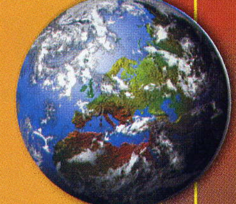
Se puede encontrar más información técnica sobre el Address Resolution Protocol en las RFC 826 y 903, que se pueden encontrar en muchos sitios de Internet, por ejemplo en www.faqs.org/rfcs/rfc826.html y www.faqs.org/rfcs/rfc903.html, respectivamente. Desgraciadamente, aún no han sido traducidas al castellano.

Derivación de una conexión con spoofing de las tablas ARP

Tráfico normal

A





Enrutador

Internet

B

Tráfico tras haber modificado la caché de ARP suplantando la dirección MAC del enrutador con la de C

A

C



Router

Internet

B

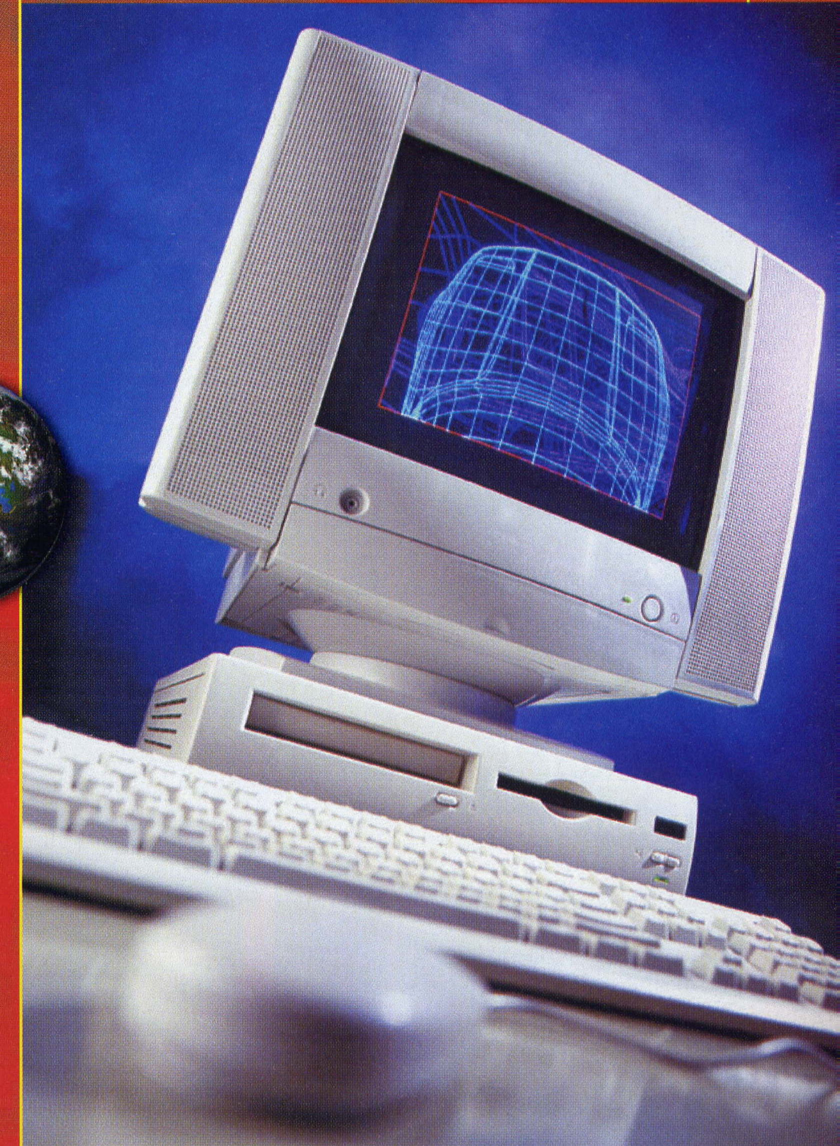
En la práctica:

./arp eth0 aa:aa:aa:aa:aa:aa ff:ff:ff:ff:ff:ff 1 MAC-NUESTRO IPROUTER 00:00:00:00:00:00 1.1.1.1 1000000

Con este paquete la caché de toda la red local se actualizará. Probablemente, el atacante enviará el paquete muy frecuentemente, probablemente cada segundo, para asegurarse de que el enrutador no envíe paquetes reply correctos o modifique la caché. Puesto que la caché se actualiza cada segundo, A y B no enviarán nunca la consulta ARP. Utilizando request en lugar de reply, el atacante probablemente pasará desapercibido, sin dejar ningún rastro evidente del ataque.

>> Cómo defenderse

Todo esto funciona sólo dentro de una red local, pero es un ataque muy eficaz y no debe ser infravalorado. Todos los equipos de la red local son vulnerables a este tipo de ataques. Además, el ataque puede llevarse a cabo remotamente si el atacante entra en la red local a través de un túnel VPN. Una primera contramedida de defensa es imponer la entrada ARP como estática. La segunda es monitorizar los ARP, por ejemplo implementando controles en los dst IP. Ello quiere decir que por cada equipo de la red que el atacante quiera desviar tendrá que enviar un arp. Por ello enviará paquetes a intervalos de un segundo; por cien equipos tendrá que enviar cien paquetes ARP por segundo en lugar de 1. Y así el ataque debería ser más evidente.

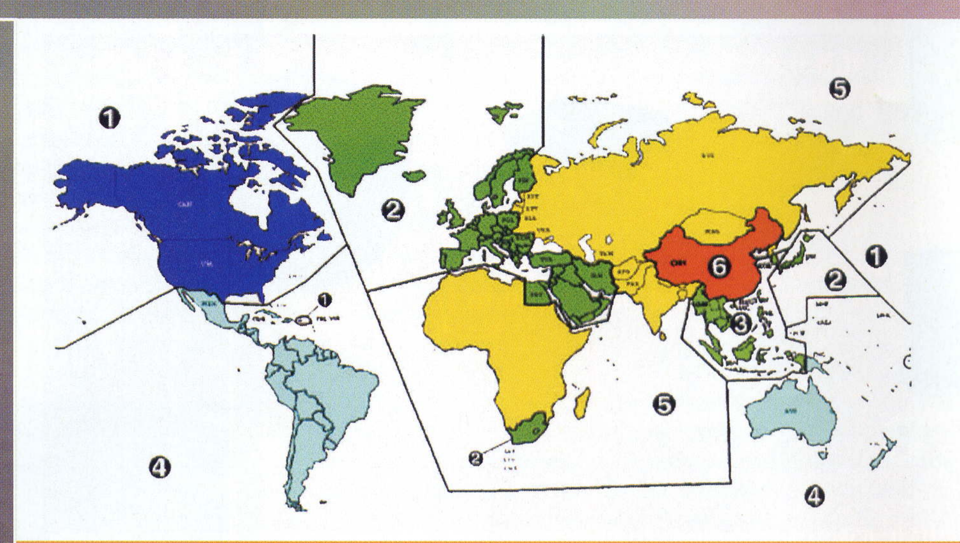


¡VIDEOLECTORES DEL MUNDO, UNIOS!

Las majors decidieron que determinada película pueda ser vista por un americano pero no por un europeo, un africano o un indio. Si no os gusta este estado de cosas, y queréis forzar este "embargo cultural", ¡seguid leyendo!

De todas las limitaciones en el uso de un DVD Video impuestas por las productoras, la más odiosa y aparentemente insulsa es la que atañe a la zona geográfica en la que se puede ver la película. Frente a la apertura de los mercados y de la libre circulación de la cultura, **las majors de Hollywood han dividido el mundo en seis áreas gegráficas; las unidades de DVD vendidas en un área no pueden utilizar los DVD vendidos en las otras cinco zonas**. Si les preguntáis el porqué de esta elección les vais a oír gruñir algo acerca de la necesidad de frenar la piratería y cosas semejantes, pero **lo que en realidad quieren las majors es poder decidir cuando y cómo (o incluso "si") una cierta película la tienen que ver los habitantes de una determinada región**. Esto les permite, por ejemplo, proyectar una película en las salas europeas meses después de su estreno en los Estados Unidos, sin que los espectadores europeos hayan visto ya el DVD, que podrían haber comprado por correo a EEUU.

Aunque **no está prohibido comprar un lector de DVD americano, africano o japonés para ver las películas que se venden en aquellas zonas**, comprar seis lectores diferentes para poder ver cualquier DVD comprado y pagado legalmente, nos parece una exigencia demasiado cara.



Mapa del imperio digital de las productoras cinematográficas. Nunca como en este caso había adquirido tanto sentido el lema "Di-Vi-De y vencerás".

>> Como funciona la protección

De una misma película se producen DVD con un código diferente para cada región. Inicialmente, los lectores de DVD (en cuanto a la mecánica del lector) eran capaces de leer los discos de cualquier zona y los posibles controles se realizaban a posteriori (en el caso de un ordenador, del software de reproducción o del sistema operativo). Es evidente que si se trata de añadir un software, éste se puede modificar como a uno le parezca y, que -tal como ocurre con los códigos para las consolas de los videojuegos- por Internet han empezado a circular las secuencias de teclas que hay que apretar en el mando a distancia del DVD para modificar la zona (para los lectores de DVD-ROM montados en los PC, existen programas que permiten hacer lo mismo).

A fines de los años 90, en vista de lo que se avecinaba, las majors impusieron una limitación más a los fabricantes de DVD: para poder obtener del DVD Forum las claves para descifrar las filmaciones, los productores tenían que apañárselas para soportar una nueva tecnología de protección llamada RPC2, que limitase la posibilidad de modificar el área geográfica del lector en cuestión. El sistema RPC2, se hizo obligatorio a partir del 1 de enero del 2000, es un poco más sofisticado y se implementa en el mismo lector. En cada lector de DVD que utiliza este sistema hay un contador. **Cada vez que se modifica la zona, el contador va contando hasta llegar a 5 y a partir de allí la zona sólo se puede modificar** desde un centro de asistencia del fabricante. Este también tiene a su disposición un número limitado de puestas a cero del contador y una vez agotadas el DVD queda bloqueado para siempre en una zona.

>> ¡Hecha la ley, hecha la trampa!

En este caso tampoco se ha modificado el hardware del lector, sino su software. Los lectores RPC2 son iguales a los RPC1 pero con un firmware diferente (el software registrado en memorias no volátiles, necesario para el funcionamiento interno de un aparato). Por lo tanto, **si se puede sustituir el firmware para las especificaciones RPC2 por el precedente RPC1, se puede tener un lector de DVD que no se bloquea después de cinco cambios de zona**. Por supuesto, sustituir el firmware RPC2 con la versión anterior RPC1 es posible y no muy complicado. En Internet hay colecciones de firmware viejos pero menos de-

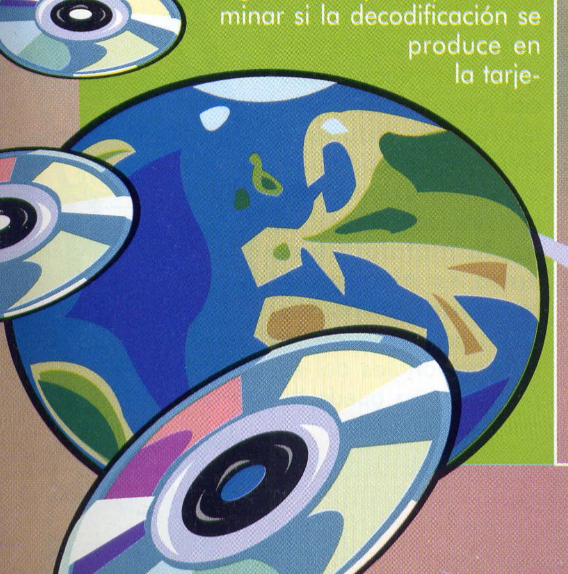


licados que los actuales en cuanto al tema del cambio de zona. Hay firmware que hacen que el lector se convierta en Region Free (sin más modificaciones de zona y por lo tanto, capaces de leer DVD de cualquier región), y otros que permiten modificar manualmente la zona sin las limitaciones impuestas por los contadores.
Lo más complicado es especificar la marca y el modelo exacto del propio lector de DVD (no basta con mirar la caja, pues a menudo el mismo modelo se vende con marcas diferentes) y encontrar la versión apropiada del firmware.

>> Manos a la obra

Antes de que os dispongáis a hacer algo con vuestro lector de DVD, os queremos advertir que **modificar el firmware es siempre una operación delicada**. Si algo va mal durante la operación (si se corta la corriente, si se cuelga el ordenador o si el firmware descargado de Internet no fuese correcto), **el lector de DVD podría quedar inutilizable**, y solo el centro de asistencia podría repararlo. Además modificar el firmware **invalida la garantía**. Si el lector lo habéis comprado hace poco, pensadlo dos veces antes de renunciar a un largo periodo de garantía y asistencia gratuita.

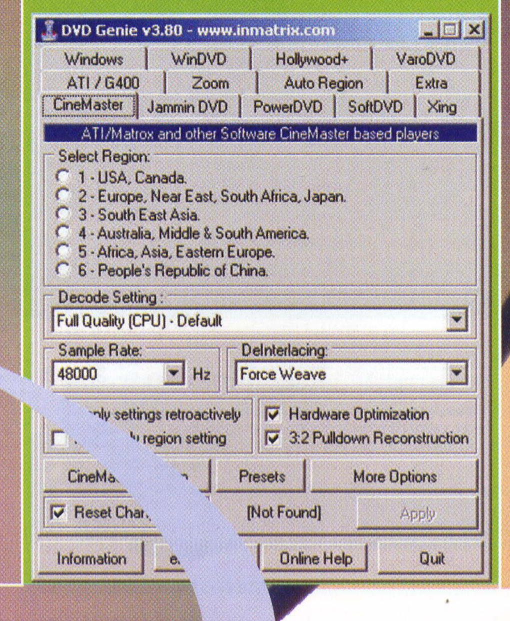
1 1. En primer lugar hay que **obtener cierta información del propio lector de DVD**. Esta operación se puede llevar a cabo con un programa como CDVDInfo (**http://digital-digest.com/dvd/downloads/cdvdinfo.html**). Si el programa percibe que el lector es Region Free, queda por determinar si la decodificación se produce en la tarjeta de vídeo (lector hardware) o si la lleva a cabo un programa (vía software). Si el lector está en una zona específica, **pasad al punto 4**.





2 **2. Si la decodificación se realiza mediante software**, la región programada se puede puentear usando programas como DvdGenie (**www.inmatrix.com**) o Dvd Region Killer (**http://digital-digest.com/dvd/downloads/dvdrk.html**), poniendo simplemente la región en 0.

3 3. Si la decodificación se hace en el hardware y el disco es Region Free, **hay que usar programas que permitan cambiar las posiciones de zona en la tarjeta de vídeo sin aumentar el contador**. Estos programas varían según el modelo de tarjeta de vídeo y se encuentran en la dirección **http://digital-digest.com/dvd/articles/region_hardware.html**. Antes de utilizar un DVD, se tendrá que controlar la región para la cual ha sido codificado y modificar la programación usando el software más apropiado.



? **Flashare**: modificar un firmware, que normalmente está escrito en un tipo de memoria definida como "memoria flash".

4 4. Si el lector no es Region Free hay que hacer que lo sea. Después de comprobar la marca y el modelo exactos id a la dirección **http://digital-digest.com/dvd/downloads/firmware.html** o al sitio **www.firmware.fr.st** y descargad el archivo correspondiente. Para flashear (instalar) el firmware, **podría ser necesario también descargar una utilidad que se os indicará en la página en cuestión**. No todos los lectores se pueden dejar libres de región. Si el vuestro no estuviese en la lista, no hay soluciones fáciles (se tendría que copiar cada uno de los soportes DVD eliminando el código regional).

EL MAC Y EL REGION CODE

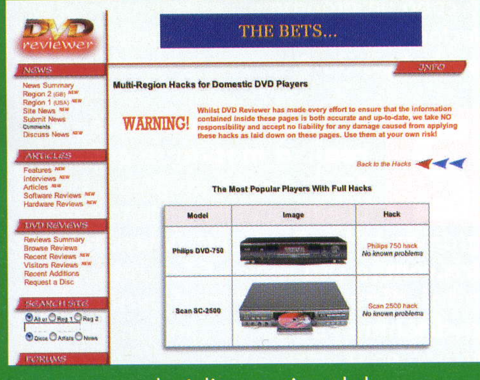
Todo lo dicho referido a Windows valen también para Mac, con una obvia excepción, los programas citados son sólo para Windows. Tranquilos, para Mac (Clásico y OS X) existe todo lo necesario para liberar vuestro lector de DVD. Todas las informaciones, los instrumentos necesarios y los patchs para el firmware de los lectores de DVD se pueden encontrar fácilmente en Internet. Uno de los sitios mejores es The Mac DVD Resource, que se encuentra en la dirección: www.wormintheapple.gr/macdvd.

5. Después de descargar el firmware y, de ser necesario, el programa para su instalación, proceded a la actualización (mejor dicho al envejecimiento), tenendo cuidado y **trabajando en condiciones de máxima seguridad**. Leed todas las instrucciones que encontréis, salid de todos los programas, y si hay tormenta puede que sea mejor retrasar la operación. **Un contratiempo en este momento inutilizaría vuestro lector.**

6. Después de haber hecho los cambios, ya sea en el software o en el hardware, es conveniente hacer que Windows reconozca nuevamente el lector de DVD (de lo contrario conservaría lo que tenía registrado antes, impidiendo la visualización de DVD de zonas diversas). Si todo ha ido bien, ahora ya podéis ver vuestra película preferida independientemente de la zona donde fue producida.

7. Os recordamos de nuevo que la operación es perfectamente legal, pero que anula la garantía del lector.

PARA LOS LECTORES DE SOBREMESA



Con los lectores de DVD de sobremesa (los que se conectan al televisor como un aparato de vídeo) evidentemente no es posible ponerse a hacer experimentos con programas parta entender la información sobre la mecánica y si fuese necesario flashear el firmware, pero no todo está perdido. Para ahorrar, los lectores de DVD de sobremesa están todos producidos de la misma manera y luego se programa el código regional de una manera u otra. En muchos casos, pues, se puede ir modificando la posición regional e incluso dejar el lector Region Free. A veces puede ser necesario abrir el lector y accionar un interruptor o un jumper, muy a menudo basta teclear una determinada combinación de teclas en el mando a distancia. En algunos casos no hay nada que hacer y la operación sólo se puede hacer en un centro de asistencia. La situación varía de unas marcas a otras y de unos modelos a otros. Normalmente Internet resulta ser una fuente interminable de información. Aquí tenéis un par de sitios obligados:

VCDHelp - DVD Player Hacks

www.vcdhelp.com/dvdplayershack.php
Una lista bastante importante de modelos, pero los trucos no siempre funcionan. Usadlos con cuidado.

DVD Reviewer - Multi Region Hacks

www.dvd.reviewer.co.uk/info/multiregion
Al ser un sitio inglés tiene el inconveniente de tener en la lista muchos lectores con las siglas con las que se venden en Europa (a veces diferentes de las de los mismos lectores en EE.UU.).



GRAGEAS

RCE, EL ÚLTIMO INVENTO DE LAS MAJORS

Sabiendo que los lectores se pueden trucar (y que en ciertos casos ya se venden como Region Free, por ejemplo en **www.zonefreedvd.com**), la Motion Picture Association of American (MPAA) ha desarrollado **un nuevo sistema, llamado RCE (Regional Code Enhacing) que impide de la visualización del DVD Vídeo en los lectores Region Free**. La única manera de evitar la protección es poner la zona antes de colocar el DVD, pero esto obviamente sólo se puede hacer con los lectores RPC-1. De lo contrario, después de cinco cambios de zona el lector se bloqueará en la última región programada. Actualmente el RCE está implementado **sólo en los DVD producidos para la zona 1 (EEUU)** , y no se sabe si se aplicará a otras regiones.

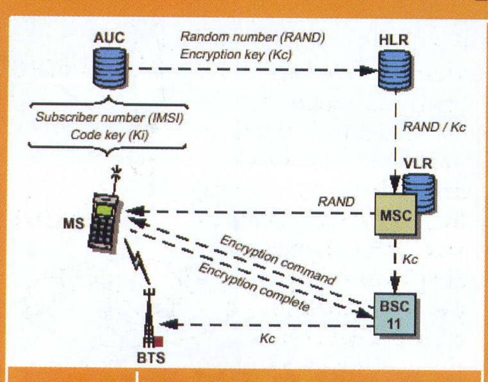
AREA CERO EN EL ESPACIO

Hace ya un par de años que en la estación espacial internacional se instalaron dos lectores de DVD con el código de área cero. Dando vueltas alrededor de la tierra, **la plataforma no entra en ninguna de las zonas previstas.** La larga mano de las multinacionales del espectáculo no puede llegar allí arriba.



¿ALGUIEN SE HACE AL GSM?

Mientras esperamos el nuevo método de codificación y cifrado del sistema de comunicación GSM, veamos cómo funciona el sistema actual y cómo se puede violar



Al parecer, los teléfonos GSM se volverán más seguros gracias a la introducción de un nuevo sistema de cifrado llamado A5/3. Se trata de una variante del anterior sistema A5, que utiliza tres algoritmos diferentes para codificar la conversación, de manera que no pueda ser oída por terceros (claro está, excluyendo la policía). La introducción de un nuevo estándar de cifrado **confirma implícitamente que el sistema precedente tiene carencias**.

La red GSM es digital, lo cual permite convertir los sonidos en bits, que **no tienen ningún significado si se "escuchan" desde una simple radio**. De hecho, no se obtiene una señal audible ni aún sintonizándolo en un canal utilizado por el sistema GSM y realizando la conversión inversa, de digital a analógica. Por un solo canal de radio se transmiten en una secuencia rápida las comunicaciones de 8 usuarios diferentes y cada conversación cambia de frecuencia continuamente. Aún suponiendo que lográsemos seguir una conversación en su continuo periplo por los canales, todavía quedaría por resolver el problema del cifrado.

Antes de ser transmitidos, los datos se cifran utilizando las claves que hay en cada tarjeta SIM (el IMSI, International Mobile Subscriber) y en el sistema de células del operador. La clave de la SIM es siempre la misma, pero la del operador se modifica frecuentemente.

Mientras que la comunicación entre teléfono y célula es bastante difícil de violar, esto no es así entre la célula y la central telefónica. En la mayoría de casos esta comunicación se hace en claro por vías terrestres o con puentes de microonda. **Esta vulnerabilidad ya la demostró un investigador** que gano 100.000 Marcos alemanes al conseguir realizar una llamada a cuenta de otro número.

» Intercepciones en caja

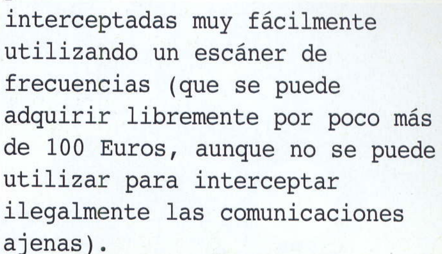
La empresa americana Gcom Technologies (www.gcomtech.com) produce un aparato capaz de interceptar todas las comunicaciones de un determinado abonado, efectuando un ataque del tipo "hombre en el medio".

El aparato es una estación base GSM puesta en una caja portátil, que **hace creer al teléfono GSM que es la célula del operador y al operador que es el abonado al que se quiere interceptar, presentándose con sus credenciales**. Ni el abonado ni el operador notan nada extraño, pero todas las llamadas enviadas o recibidas las graba la caja infernal. El sistema sólo se vende a representantes de gobiernos o de las fuerzas policiales previa acreditación, pero no es improbable que alguien pueda hacerse con una, igual como ocurre con las armas.



Estación base: Son las estaciones que comunican con los teléfonos y retransmiten a la central en la red GSM de un operador.

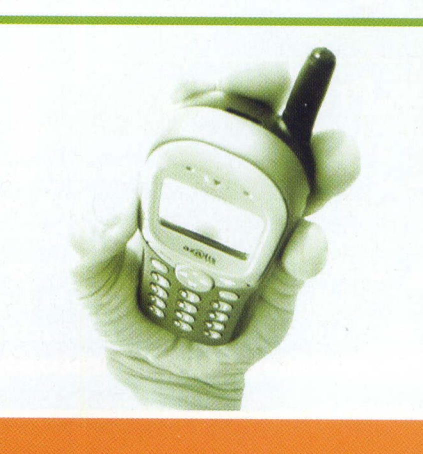
PRATICA



TRASFORMAR EL MÓVIL EN UN ESCÁNER

Decíamos que las comunicaciones ETACS pueden ser interceptadas muy fácilmente utilizando un escáner de frecuencias (que se puede adquirir libremente por poco más de 100 Euros, aunque no se puede utilizar para interceptar ilegalmente las comunicaciones ajenas).

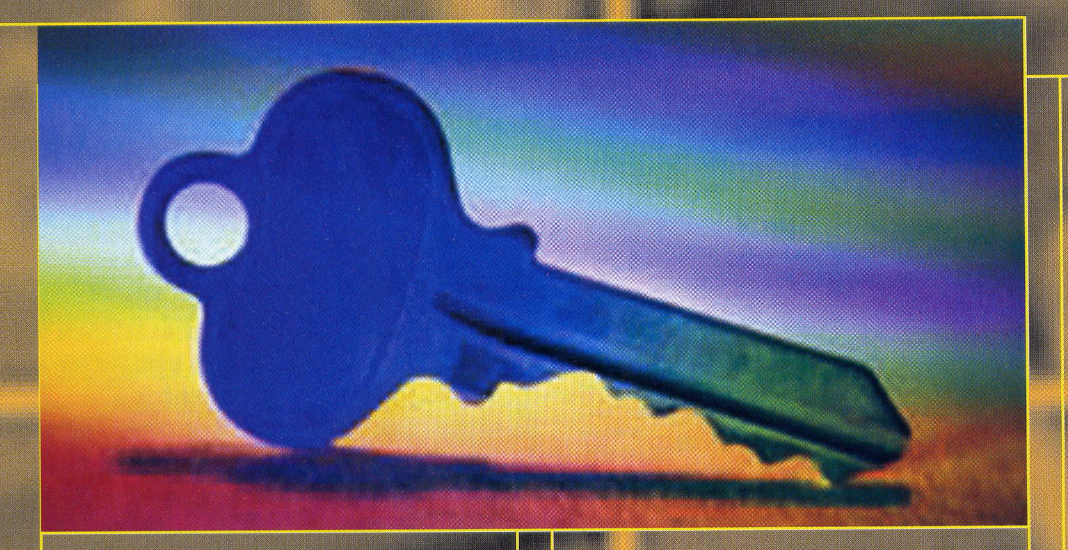
Algunos teléfonos celulares ETACS pueden ser fácilmente transformados en un escáner, dado que disponen de un receptor-transmisor que funciona con las frecuencias adecuadas. Por ejemplo, con algunos modelos de ETACS de Motorola basta con abrir el compartimento de la batería, colocar un trozo de papel de aluminio en el contacto central (normalmente más bajo que el contacto presente al otro lado), y volver a encender el teléfono que ahora entra en la modalidad de Test. Tecleando # aparece escrito Tac5 en el display. Ahora basta escribir 08 y luego nuevamente # y el teléfono se habrá transformado en un escáner. Desde el teclado se podrá marcar un canal de radio comprendido entre 1101 y 1199, y confirmando con # se tendrá acceso a las llamadas efectuadas en la célula en la que se encuentra.



CÓMO FUNCIONAN LAS CONEXIONES CIFRADAS VÍA WEB, TELNET O FTP

Los secretos de SSL

Volvamos a examinar Secure Socket Layer, uno de los protocolos de cifrado más difundidos por la web y las conexiones Telnet.



Vamos a buscar entre los recovecos de Secure Socket Layer para entender su funcionamiento hasta el mínimo detalle.

El protocolo SSL (Secure Socket Layer) es un protocolo de Netscape, pero lo soportan también otros navegadores. Garantiza la privacidad de las comunicaciones en Internet y permite la comunicación entre cliente y servidor de una manera segura y privada.

De hecho, este protocolo se utiliza mucho en conexiones en las cuales hay que enviar información confidencial, como por ejemplo el número de una tarjeta de crédito, nombres de usuario o contraseñas para acceder a sitios o servicios protegidos.

>> Privacidad

Son muchos los aspectos que hacen que este protocolo sea seguro y fiable. Vale la pena examinarlos en detalle.

El sistema de cifrado parte de después del HandShake hasta el final de la conexión ya que incluso los datos enviados se cifran y por ello se utiliza el encriptado simétrico (DES y RC4).

DES: es el acrónimo de Digital Encryption Standard, un algoritmo de cifrado que utiliza claves de 64 bits y no tiene una elevada potencia de cálculo con respecto a las actuales. A pesar de ello, este algoritmo se utiliza mucho y a menudo en su variante Triple-DES, basada en el uso de DES repetido tres veces.

Con este algoritmo, el texto sin cifrar en el input y el texto cifrado en la salida tienen una longitud estándar de 8 bytes. El input tiene pues que ser múltiple de este bloque elemental. Si la longitud del mensaje no corresponde a un múltiple de esta medida debe rellenarse con datos hasta llegar a la medida necesaria para operar en modo CBC o EBC correctamente.

La clave de cifrado está formada por 56 bits aleatorios y 8 bits iguales, que compondrán la clave de 64 bits.

3DES: Este método, soportado por muchos sistemas, es hijo del anterior y consiste en la ejecución del DES tres veces consecutivas para triplicar el número de bits en la clave de cifrado. Esta técnica se conoce como EDE (Encrypt-Decrypt-Encrypt).

El proceso de decodificación puede hacerse compatible con el precedente, deteniendo el mecanismo a la mitad. Y si las tres claves utilizadas son las mismas, el triple DES es equivalente a un solo cifrado DES. Así, una aplicación que puede usar sólo el DES es capaz de comunicar con otra que esté usando el triple DES. En cambio, si las tres claves son diferentes el segundo desencriptado obstaculizará al mensaje opuesto y éste no podrá descifrar el primer estado.

RC4: Es un algoritmo de la RSA Data Security, Inc. Originariamente, sus especificaciones en la fase de proyecto se consideraron secretas, pero en 1994 fueron divulgadas.

Es un algoritmo que se usa mucho en varios tipos de aplicaciones.

Utiliza la clave que le proporcionan los usuarios para producir una secuencia numérica seudo-casual, ligada a una operación de O exclusivo (XOR) con los datos del input. Ello significa que las operaciones de encriptado y de desencriptado son idénticas. El número de bits de la clave es variable, y abarca desde un mínimo de 8 a un máximo de 2048; el código que utiliza tiene una longitud diez veces inferior al DES (con una menor seguridad). Pero la ventaja estriba en la velocidad de ejecución (casi cinco veces más rápida). No se le conocen ataques. La versión internacional del RC4 a 40 bits fue violada con el método de la fuerza bruta en ocho días por dos asociaciones.

>> Autenticación

Este proceso se realiza utilizando sistemas de cifrado asimétrico o de clave




pública como RSA y DSS. De este modo estamos seguros de comunicar directamente con el servidor correcto (la autenticación la piden tanto el servidor como el cliente).

RC4: Es el acrónimo de Rivest Shamir Adelman y se considera muy seguro si se usan claves largas de entre 768 y 1024 bits. Este algoritmo de clave pública es el más utilizado tanto para cifrar como para las firmas digitales. Su funcionamiento es similar a lo siguiente:

1. A genera 2 números primos grandes p y q
2. A calcula n = p ◊ q y f(n) = (p - 1)
3. A elige un número 1 < y < f(n) de modo que gcd(e, f(n)) = 1
4. A calcula d = e-1 mod f(n) usando el algoritmo de Euclides extendido
5. A publica n y e como su clave pública PA = (e, n).
6. A conserva n y d como su clave privada SA = (d, n).

DSS: Es el acrónimo de Digital Signature Standard. No es muy fiable y sólo se utiliza para la firma y todavía no se ha hecho del todo público.

>> Multiplataforma

SSL es multiplataforma y trabaja tanto en Windows como en Solaris.

Hace algún tiempo, el gobierno americano imponía duras limitaciones a la utilización de las técnicas de encriptado "fuertes", de modo que no se podían utilizar claves más largas de 40 bits . Hoy estas limitaciones ya no existen y por fin se pueden descargar y utilizar legítimamente navegadores que soportan las claves largas.

HandShake y análisis de los procesos

Ante todo, los protocolos utilizados durante la secuencia de handshake son:

"SSL Handshake Protocol" para establecer una sesión entre el cliente y el servidor

"SSL Change Cipher Spec protocol" para concordar la Cipher Suite para la sesión.

"SSL Alert Protocol" para comunicar mensajes de error SSL entre cliente y servidor. Veamos cómo funciona una conexión (cliente -> servidor) con SSL

En la primera parte el cliente y el servidor concuerdan la versión del protocolo y los algoritmos de encriptado que se van a usar, y luego usan el cifrado en clave pública para intercambiar los datos encriptados.

Veámoslo esto en detalle: el cliente envía al servidor un Hello y éste responde de la misma manera (con un Server Hello). Esto tiene un valor importante pues es durante esta fase cuando se establece la versión del protocolo, la Cipher Suite, el ID de sesión y el método de compresión.



Si durante esta fase algo falla o cae, la conexión se interrumpe. Si no hay ningún problema con la conexión, el servidor manda un mensaje de Server Hello Done para indicar al cliente que la fase hello message del HandShake ha terminado con éxito y espera una respuesta positiva del cliente.

En este momento intercambiaran los datos que hemos mencionado antes.

La fase de HandShake ha terminado y durante la conexión el servidor puede mandar múltiples Hello Request aunque estos serán ignorados por el cliente.

Y, al contrario, el cliente puede mandar a su vez Client Hello para renegociar los datos de una conexión preexistente. Un completo ejemplo de HandShake es este:

```
Cliente ------> Client Hello ------> Servidor
Cliente <------ Server Hello <------ Servidor

Cliente <------ Certificate <------ Servidor
Cliente <------ Certificate Request <------ Servidor
Cliente <------ Server Hello Done <------ Servidor

Cliente ------> Certificate ------> Servidor
Cliente ------> Certificate Verify ------> Servidor
Cliente ------> Change Chiperspec ------> Servidor
Cliente ------> Finished ------> Servidor

Cliente <------ Change Chiperspec <------ Servidor
Cliente <------ Finished <------ Servidor
```

El Client Hello tiene una estructura como la siguiente:

```
struct {
ProtocolVersion client_version;
Random random;
SessionID session_id;
CipherSuite cipher_suites<2..215>;
Compression Method compression_methods<1..27>;
} ClientHello;
```

mientras el servidor sólo tiene una estructura como ésta:

```
struct {
ProtocolVersion server_version;
Random random;
SessionID session_id;
CipherSuite cipher_suite;
CompressionMethod compression_method;
} ServerHello;
```

server_version: contiene la versión del protocolo

Random: es una estructura totalmente casual generada por el servidor que no tiene ninguna dependencia del Mensaje Hello del cliente.

Compression_method: el método de compresión dado por el servidor.

Una chiper suite se define por tres componentes:
- Método de intercambio de la clave
- Algoritmo de cifrado para la transferencia de datos

CÓMO FUNCIONAN LAS CONEXIONES CIFRADAS VÍA WEB, TELNET O FTP

- Message Digest para la creación del MAC (Message Autentication Code)

1. Método de intercambio de la clave: El método de intercambio de la clave sirve para definir como se acordará a continuación la clave secreta. SSL 2.0 soporta sólo el intercambio de claves RSA, mientras la versión sucesiva SSL 3.0 soporta varios algoritmos de intercambio.

2. Algoritmo de cifrado para la transferencia de datos



Para el cifrado de los datos, SSL usa algoritmos de encriptado simétrico. Se pueden llevar a cabo ocho elecciones:

Cifrado Bloques
.RC4 con clave de 40-bits
.RC4 con clave de 128-bits

CBC Cifrado Bloques
.RC2 con clave de 40-bits
.DES40, DES, 3DES_EDE
.Idea
.Fortezza

También cabe la posibilidad de no seguir ningún cifrado.

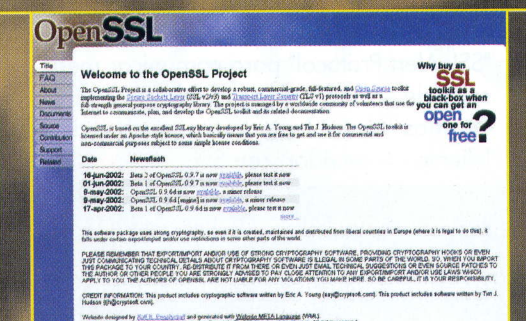
3. Message Digest para la creación del MAC

Éste determina cómo se creará la huella digital del registro.
Las posibilidades son tres:

MD5, con hash a 128-bits
SHA (Secure Hash Algorithm) con hash a 160-bits

O aquí también se puede evitar de elegir una huella.

Server certificate: para una mayor seguridad, durante el HandShake el servidor envía el Server Certificate o bien el certificado que se envía inmediatamente después del Server Hello.
Si el servidor no dispone de un certificado manda un mensaje de Server Key Exchange. El servidor podría incluso pedir un Certificate Request, o bien un Client



En la dirección www.openssl.org se encuentran las especificaciones y las fuentes de OpenSSL, una implementación Open Source de SSL.

Certificate (aunque no sucede frecuentemente).

Client certificate: el cliente después de recibir un Server Hello Done manda su certificado.

Secret Premaster message (RSA): el cliente genera un mensaje premaster de 48 bytes usando el algoritmo de clave pública del servidor que tiene una estructura como la siguiente:

```
struct {
ProtocolVersion client_version;
opaque random[46];
} PreMasterSecret;
```

Client_version y random ya los hemos visto antes.

Pre_Master_Secret : es el valor generado al azar por el cliente utilizado para generar el verdadero Master Secret

```
struct {
public-key-encrypted    PreMasterSecret
pre_master_secret;
} EncryptedPreMasterSecret;
```

>> ¿SSL=seguridad?

En teoría, la respuesta tendría que ser SÍ, pero en la práctica es muy diferente. De hecho SSL cuenta con varios fallos y puede ser "fácilmente" (es un decir) violable con medios como:
Criptoanálisis
Fuerza Bruta
Replay

El uso de RC4 con claves de 40 bits puede que parezca algo poco seguro, y de hecho es así.
En España es obligado por la ley de Estados Unidos sobre la exportación de los algoritmos de encriptado.
Se han descubierto muchos otros errores en este protocolo y como acostumbra a suceder, bugtraq es un óptimo instrumento para estar al día.
Antes de dejar el tema, apuntemos dos últimas cosas muy importantes.

1. El protocolo SSL no es un protocolo independiente sino que se basa en otro protocolo, el TCP/IP.

2. SSL se aplica a muchos servicios utilizados en todas partes y muy frecuentemente, como telnet y ftp:

SSL-telnet:
ftp://ftp.psy.uq.oz.au/pub/Crypto/SSLapps/

SSL-ftp:
ftp://ftp.psy.uq.oz.au/pub/Crypto/SSLapps/



www.eviltime.com





Andrei, de profesión... ¡pirata!
Nombre: xxxxxx
Edad: 24 años
Nacionalidad: ucraniano
Cabello: rubio
Ojos: azules
Profesión: pirata



El pirata ucraniano

Después de haberse movido por la escena rusa durante un par de años, donde se divertía penetrando (o al menos intentándolo) en los sistemas informáticos americanos y alemanes, Andrei se encontró en medio del cracking "profesional" por medio de sus amigos del ambiente underground. Su objetivo: craquear programas y juegos para ponerlos a la venta en versión pirata.

¿Qué es lo que te motiva en el hacking/cracking?

El hecho de poder ganar un centenar de dólares. Porque muy a menudo sólo necesitamos un editor hexadecimal y algún software desarrollado en C que nos permite comparar el código para poder en unas horas desenmascarar cualquier programa ¡En el 85% de los casos!

¿Concretamente, cómo funciona?

Todo varía mucho en función de los editores de programas. La mayoría de las veces conseguimos tener las versiones americanas en el momento en que salen e inmediatamente hacemos una copia de ellas modificando el código. Y otras veces, menos frecuentemente, desarrollamos un patch corrector para ejecutar después de la instalación del programa o del juego. La última posibilidad es colocar un número de licencia con la versión que nosotros distribuimos.

¿No tenéis problemas con la policía?

Digamos que hay que saber nadar y guardar la ropa. Intentar disuadir a la policía mediante una generosa donación para sus "obras de caridad". ¿No sé si me entiendes...? Pero sobre este último punto no somos nosotros quien tiene que responder sino nuestro jefe.

¿Quieres decir que trabajas para alguien?

Claro, aquí no se puede hacer nada a un cierto nivel sin tener unas bases férreas. A mi nivel, no represento nada y tampoco intento tirar adelante yo solo, es demasiado arriesgado.

¿Te estás refiriendo a la mafia?

¡No nos pasemos! Se trata de grupos de interés común que no quieren forzosamente repartirse el pastel, algo que puedo entender.

¿Y por qué no trabajar en una empresa de informática clásica, por ejemplo, en el Oeste?

No tengo ganas de trabajar diez horas al día por 250 dólares (225 Euros) cuando actualmente gano más de 800 dólares por unas horas a la semana. En cuanto a irme al Oeste, la verdad, ¡aquí tengo todo lo que me hace falta!

¿Cuántos trabajáis i cómo?

Somos un pequeño equipo de cuatro personas, más una persona externa que trabaja como freelance según nuestras necesidades. Somos un programador de software y patches, dos programadores/crackers que se dedican a la protección, y un diseñador para las solapas y la fabricación de los CD, que sirven para la duplicación. A veces, y cada vez es más frecuente, tenemos un desarrollador multimedia que desarrolla la interfaz de presentación de los CD piratas (los descendientes de los demomakers que creaban las demos adjuntas con los softwares piratas).

¿Cómo veis vuestro futuro? ¿Tenéis confianza?

Sabéis, el futuro se limita al mes que viene. Yo me gano bien la vida, no me arriesgo más de lo necesario y ya veré qué hacer cuando sea el momento.

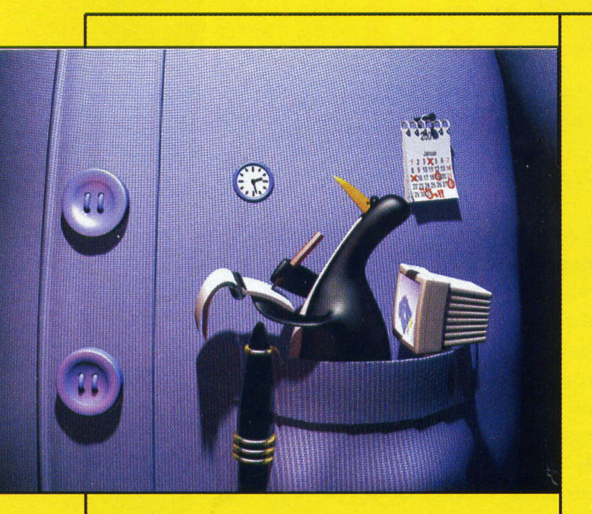
Aquí no nos llega ningún CD de Europa del Este, cuando podría suponer una enorme cantidad de dinero, ¿por qué?

No conozco los secretos de los jefes, sé que es un tema que interesa, pero por el momento prefieren no moverlo.

TODO LO QUE TE GUSTARÍA SABER SOBRE LINUX...

Linux en pelotas

Después de haber presentado las más famosas y representativas distribuciones de Linux nos disponemos a hacer una pequeña lista de preguntas más frecuentes que los novatos a menudo se hacen: las bases técnicas, la filosofía que hay detrás del código abierto o el mercado. ¡Empezamos ya!



1 ¿Qué diablos es Linux?

Linux es un **sistema operativo para ordenadores PC** (386-Pentium PRO, Digital Alpha, PowerPC, Sun SPARC, Apple Macintosh, Atari ST/TT, Amiga, MIPS) desarrollado como implementación gratuita de UNIX. Sus primeras versiones las hizo Linus Torvalds en la Universidad de Helsinki en Finlandia. A partir de aquí y gracias a su estructura de código abierto, muchos desarrolladores y programadores de todo el mundo contribuyeron de una manera determinante al progreso de este sistema operativo.

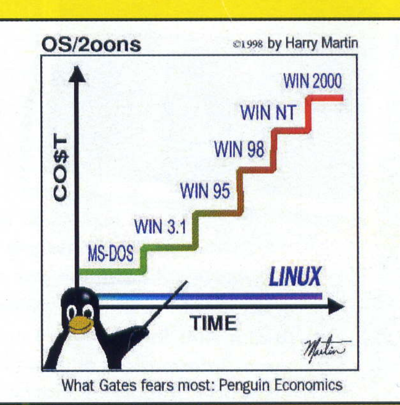
2 ¿Por qué todo el mundo dice que es mejor que Windows?

Linux nace y se desarrolla a lo largo de los años gracias a la **contribución de muchísimos desarrolladores**, que siguen mejorando sus características y lo hacen día a día más fiable y seguro. La filosofía que hay detrás de un proyecto de código abierto está por naturaleza consagrada al perfeccionamiento constante del producto. Por el hecho de mantener el código visible y que cualquiera lo pueda modificar está constantemente sujeto a mejoras. Un desarrollo de este tipo es conceptualmente superior (al menos desde un punto de vista técnico) al clásico sistema comercial con una fase de investigación, una fase de desarrollo y una fase de solución de bugs, que en la mayoría de los casos no hace más que dañar al usuario final, ya sea un particular o una empresa.

Las dotes de seguridad, **estabilidad y robustez de Linux** las ha heredado directamente de la compatibilidad de UNIX y de su proverbial versatilidad. Especialmente en lo que respecta a sistemas multiusuario como los que hay en red, se mantiene y se hace cada vez más eficaz.

3 ¿Pero, qué es exactamente este código abierto?

Código abierto no quiere decir simplemente tener el código fuente de un programa a nuestra disposición. Hay ciertos criterios que un producto de código abierto tiene que satisfacer (podemos hablar de un producto, puesto que no sólo los programas para ordenador pueden ser de código abierto). Si tomamos como ejemplo el software, éste debe poderse **distribuir libremente, sin pagos a cambio o limitaciones del uso**. Tiene que permitir al usuario final, como ya sabemos, **acceder al código fuente** del software en cuestión. También tiene que ser posible **efectuar modificaciones a dicho código**. Además existen muchos otros criterios más burocráticos, como por ejemplo preservar la integridad del código fuente del autor del programa en cuestión y múltiples otras cláusulas acerca de la política actual para licenciar los diversos programas. Encontraréis un buen resumen del tema en: **http://www.apogeonline.com/openpress/op_definition.html**



What Gates fears most: Penguin Economics

4 Pasando a temas más prácticos, ¿qué tengo qué hacer para tener el tal Linux?

La práctica más simple y más común para un novato que se introduce por primera vez en el mundo de Linux es sin duda la de buscar **alguna revista en el quiosco que lo distribuya**. Muchas veces se distribuye de esta manera, prácticamente gratis, para aumentar su difusión en la mayor medida posible y obtener in-

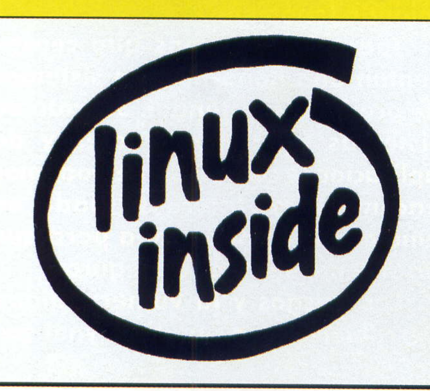



gresos en el momento en el que el cliente, satisfecho con el sistema operativo, pida, por ejemplo, asistencia sobre productos o paquetes presentes en una determinada distribución.

Si se dispone de una conexión a Internet suficientemente rápida, **muchas de las más famosas distribuciones Linux de pueden descargar gratuitamente**. Una vez ya tenemos nuestra distribución (para el que empieza aconsejamos versiones tipo Mandrake o RedHat) lo tendremos que instalar. Una vez creada una partición vacía o por lo menos que pueda alojar Linux (operación que en algunas distribuciones se produce en la fase de instalación), sólo tendremos que insertar el primer CD-ROM en el lector e iniciar la máquina. En breve habremos instalado una Linux Box completa con todo lo que nos hace falta para trabajar.

5 ¿Así puedo instalarlo sin borrar mi partición de Windows?

Sí, basta con acordarse de crear una partición vacía o apropiada para Linux. Esto se puede hacer utilizando las herramientas que facilitan algunas "distri" en fase de instalación o mediante el uso de programas como **Partition Magic o Fips, que a menudo está incluido en los CD-ROM de las distribuciones más famosas**, y que permite en pocos pasos configurar una partición para el nuevo sistema operativo. Una vez instalado Linux este configurará algo llamado **boot-loader, un software que cuando iniciemos el ordenador nos permitirá elegir con qué sistema operativo queremos trabajar**, según nos convenga.



6 ¿Entonces, mis programas para Windows funcionan en Linux?

De modo predeterminado, no. Los programas que comúnmente se usan en Windows no pueden funcionar en Linux, puesto que estamos hablando de un sistema operativo profundamente diferente de Windows en su estructura.

Pero existen diversos emuladores mediante los cuales se pueden hacer funcionar algunos programas Windows dentro de Linux. Hay que decir que el abanico de software disponible para Linux es inmenso i a menudo podremos encontrar clones perfectos de algunos famosos programas para Windows. En algunos casos, y no pocos, la calidad y la estabilidad de los clones supera las de los programas originales.

7 He oído decir que los módems no funcionan con Linux. ¿Es cierto?

En parte sí. En realidad, son sólo algunos los módems que tienen problemas con Linux, concretamente los mal reputados winmodems. Muchísimos módems internos son winmodem. Algunos componentes de los circuitos del dispositivo de comunicación de estos módems los emula Windows para reducir el coste del propio módem y esto hace que muchos de ellos sólo sean compatibles con Windows y compañía. **Lo más fácil es conseguir un módem externo y lo podemos encontrar por pocas decenas de Euros**. Se puede, de todos modos, intentar que funcione nuestro módem interno con Linux. Para ello, un sitio de referencia excelente es **http://www.linmodems.org**

8 He instalado Linux y he configurado mi módem. Me he bajado un programa y no hay modo de instalarlo. ¿Qué hago, me pego un tiro?

¡No! En general instalar cualquier programa para Linux no es tan fácil como hacerlo en Windows & Co. No es que el asunto presente especiales dificultades, imposibles de superar, pero el proceso puede ser profundamente diferente.

Es común al ambiente Linux la utilización de paquetes comprimidos del tipo tar.gz, que son más o menos el equivalente a .zip. Una vez descomprimidos los paquetes (con cualquier distribución que implemente un escritorio gráfico la operación es bastante parecida a la que se realiza para descompactar un .zip).

Nos encontraremos frente a una serie de archivos que contienen los códigos fuente del programa que se tiene que instalar. **La praxis es la de actuar desde la línea de comandos: poner en marcha un script que verifica si el ordenador tiene todos los requisitos para instalar el software y configurarlo en consecuencia**: En el 99% de los casos se hace con un ./configure.

Una vez hecho esto (es la fase en la que se pueden encontrar los problemas más graves) habremos creado un makefile, que se utilizará con make para compilar efectivamente el programa. Escribamos ./make install e instalará el programa y lo hará disponible para los usuarios.

El proceso puede ser muy complejo y variar de programa a programa. **Algunas distribuciones ya hace tiempo que han introducido un sistema de paquetización del software que hace mucho más simple la instalación de los paquetes**. Un ejemplo es la clásica forma .rpm adoptada por Red Hat y más tarde por Mandrake.

Si el programa que buscamos está disponible en estos formatos para nuestra distribución, y no tenemos tiempo o ganas de perder una tarde intentando resolver algún problema, será mucho más fácil descargar el paquete adecuado.

No por ello tenemos que fiarnos ciegamente de que los paquetes precompilados de los programas sean la mejor elección. Como siempre el mejor consejo es curiosear, manipular y leer cuanto más mejor. La documentación es enorme, ¡usémosla!



Paquetes: Son las diversas versiones de Linux que se distribuyen más o menos gratuitamente. "Distri" para los amigos...

LEVANTAR UN MURO CONTRA LOS ATAQUES EN RED

Objetivo: Honeynet

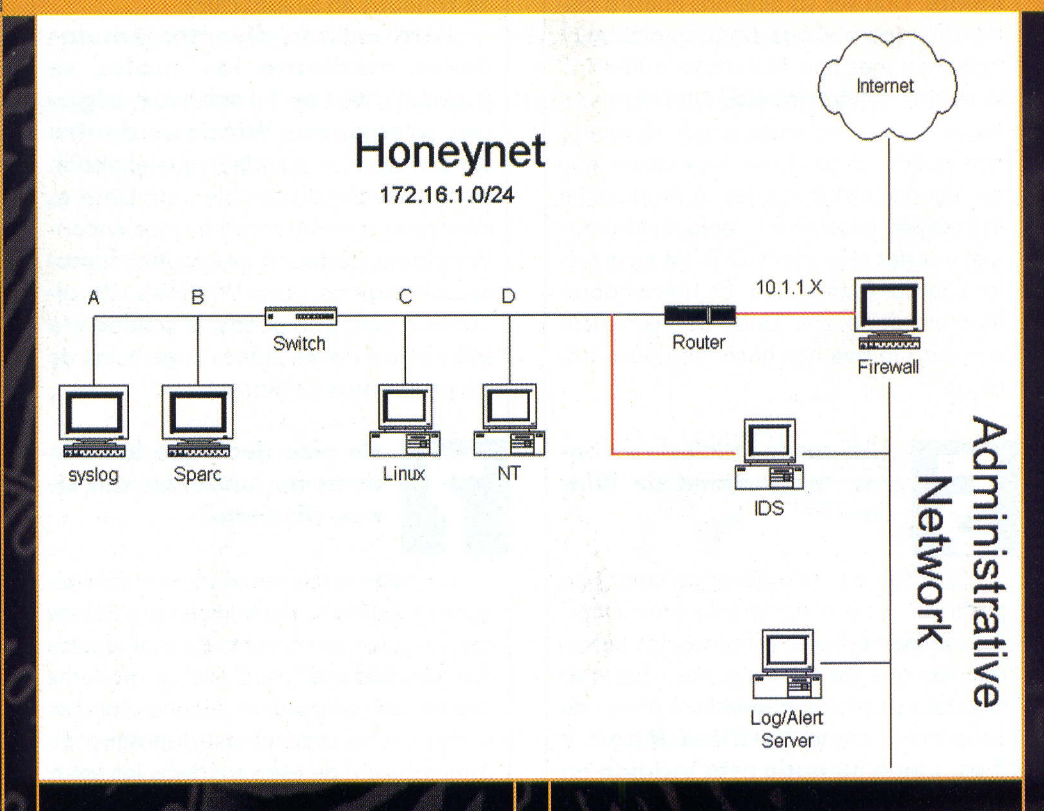
¿Qué es un honeynet? ¡Muy fácil, un honeypot! Bueno, será mejor que leáis el artículo...

The Honeynet project

En la página 30 de este mismo número entrevistamos a Lance Spitzner, el responsable del proyecto Honeynet. Pero, ¿qué es un Honeynet? Es un tipo de honeypot específico para la investigación. Un honeypot es un recurso cuyo valor es el de ser objeto de búsqueda, de ser atacado o comprometido. Su principio es el de atraer con engaño al posible attacker o demostrar los ataques.

Normalmente, son sistemas independientes que emulan a otros sistemas, o que emulan servicios conocidos o vulnerabilidades, o crean ambientes jailed. Algunos ejemplos excelentes de honeypot incluyen Specter (http://www.specter.com), Mantrap (http://www.recourse.com/product/ManTrap/), o el nuevo Deception Toolkit (http://www.all.net/dtk/). Una honeynet se diferencia de las honeypot tradicionales por su finalidad puramente investigadora. Ésta no es una solución mejor respecto a los honeypot tradicionales, sino que más bien tiene finalidades diferentes. El valor de una honeynet más que captar a un engañado o constatar un ataque es el de obtener información sobre una posible amenaza.

Hay algunas diferencias respecto a un honeypot clásico: no se trata de una sola máquina sino de una red de sistemas. Esta red se pone detrás de un dispositivo con control de acceso, normalmente un cortafuegos, donde todo el tráfico de entrada y de salida se controla y captura. Los datos capturados se van analizando para conocer las herramientas, las tácticas y los motivos de los blackhats.



Las honeynets pueden utilizar varios sistemas operativos al mismo tiempo como Solaris, Linux, Windows NT y enrutadores Cisco, los switch Alteon, etc. Esto crea un ambiente de red lo más similar posible a una red de producción.

Al disponer de diferentes sistemas operativos y aplicaciones como un DES Server en Linux, un Information Server sobre plataforma Windows, un RDBMS en Solaris, podemos conocer herramientas y tácticas diferentes.

Algunos blackhats investigan sistemas operativos específicos, aplicaciones o vulnerabilidad. Al ser variados los sistemas operativos y las aplicaciones, podemos detectar con precisión cual es la tendencia de crecimiento de un determinado fenómeno.

Todos los sistemas que representan una honeynet son sistemas de producción estándar. Se trata de sistemas operativos reales y de aplicaciones como las que podemos encontrar en Internet. Nada se emula y no se hace nada para que un sistema sea menos seguro.

Los riesgos y la vulnerabilidad descubiertos con una honeynet son





los mismas que existen en muchas empresas hoy. Es posible tomar un sistema en producción y trasladarlo a la Honeynet para conocer sus puntos débiles.

A pesar de que una honeynet puede utilizarse como un honeypot tradicional para detectar los intentos de acceso no autorizados, a menudo mantener una honeynet requiere más trabajo, riesgo y administración.

>> Requisitos principales

Hay dos requisitos principales para una honeynet: el control de los datos y la captura de los datos. Si uno de ellos falla tendremos problemas en nuestra honeynet.

El control de los datos sirve para mitigar los riesgos. Una vez que hemos pescado una máquina en la honeynet no tenemos que permitir que el attacker utilice este sistema para dañar otros recursos presentes en Internet.

Un tercer requisito podría ser el de recopilar datos, pero sólo para las organizaciones que tienen honeynets múltiples en ambientes distribuidos. En el caso de honeynets múltiples distribuidas lógicamente o físicamente por el mundo, los datos se tienen que conservar de un modo centralizado para aumentar el valor de los datos capturados.

Veamos en detalle cómo se estructura una honeynet y cómo funciona:

Como se puede ver en la figura 1, el cortafuegos separa la honeynet en tres redes, Honeynet, Internet y red de administración. Cada paquete que entra o sale de nuestra honeynet tiene que pasar por el cortafuegos y por el enrutador.

El cortafuegos representa nuestro dispositivo primario destinado al control de acceso de paquetes en entrada y salida mientras el enrutador se utiliza para reforzar este tipo de control de acceso protegiendo a nuestra honeynet de ataques de tipo spoofing, Denial of service, ICPM.

Colocar un enrutador además del cortafuegos tiene la finalidad de crear un ambiente lo más realista posible. Normalmente, en este tipo de infraestructura se tiende a limitar el número de conexiones de salida de la honeynet a 5. Esto se consigue con cualquier tipo de cortafuegos, ya sea Firewall-1 de Checkpoint, IPFILTER o IPTABLES de Linux.

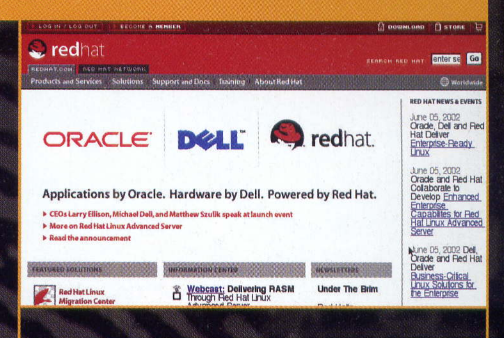
La captura de datos es el eje de toda la operativa. Para capturarlos podemos utilizar el cortafuegos o nuestro IDS.

El sistema IDS nos puede proporcionar información detallada sobre el tipo de ataque y puede grabar toda la información que pasa por la red. Un nivel más de registro lo representan los log de los mismos sistemas que se mandan a un syslog centralizado vía conexiones encriptadas.

>> Hay honeynet y honeynet

Una nueva evolución de las honeynets en este momento la representan las virtual honeynets. Se trata de un único sistema físico, que utilizando un emulador de PC como VMWARE (http://www.vmware.com) permite tener un sistema operativo real a elegir entre Linux, Openbsd, Freebsd, Solaris y Windows y cuenta con todos los requisitos de una honeynet. De ésta obtendremos un menor coste de gestión y de administración.

Otro instrumento interesantísimo de las virtual honeynet es honeyd de Niels Provos (se encuentra en el URL: http://www.citi.umich.edu/u/provos/honeyd/).



Honeyd es un daemon que crea virtual hosts en una red. Los hosts se pueden configurar para tener servicios ficticios y emular diversos sistemas operativos.

Es posible ejecutar ping y traceroute en las máquinas virtuales.

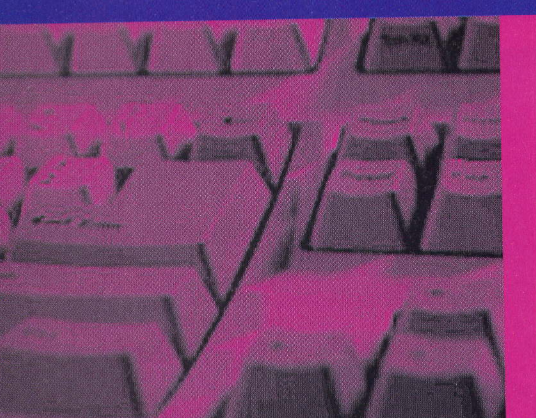
En este breve recorrido hemos visto como una honeynet puede ser un instrumento interesantísimo para investigar y conocer los puntos débiles de la propia red y para conocer nuevos ataques y vulnerabilidades. Pero hay que tener en cuenta que una honeynet es un instrumento delicado que requiere constante mantenimiento y largos períodos de análisis.

Mientras que por un lado una honeynet se puede poner en evidencia en 30 minutos, el análisis del sistema comprometido puede precisar 30 o 40 horas. En definitiva, una honeynet no resolverá nuestros problemas de seguridad, sino que los aumentará, proporcionándonos a cambio infinidad de información muy útil.



Solaris: Sistema operativo para la Red basado en Unix como Linux y desarrollado por Sun Microsystems.

Stringa: Serie de caracteres ligados, como "Utilizarinternetsinesfuerzo". Las stringas aceptan todo tipo de caracteres, tanto letras, como cifras o signos de puntuación.



Una cuestión de "protocolo"

La conexión a Internet, aunque banal, se basa en una serie de conexiones complejas... ¡Veamos cuáles!

Cuando se habla de redes de ordenadores a menudo los entusiastas, pero poco experimentadas, arrugan la nariz diciendo: "Demasiado difícil, ¡hay que ser un científico para entender estas cosas!

Es absolutamente falso, sobre todo gracias al enorme progreso tecnológico (y a las guías como esta :-P) que se ha producido en el ambiente informático estos últimos años. Gracias a ello, todo lo que antes parecía un poco más desagradable, ahora resulta bastante simple hasta para los usuarios menos expertos. No podemos dejar de hablar de la teoría que hay detrás de las redes informáticas puesto que se trata de argumentos algo más complejos que la informática. Así, el objetivo es informaros de las novedades y los cambios de las redes informáticas de todo tipo. Hablaremos de redes y de protocolos informáticos que han hecho historia y que todavía están muy extendidos.

Protocolo Ipv6 (ver eset artículo)
Protocolo TCP
Protocolo netbios/netbeui
Protocolo IPX/SPX
Protocolo AppleTalk
La Red Token-Ring
La Red Ethernet
Las tipologías de red

>> Protocolo TCP

Definir TCP/IP como protocolo no es del todo correcto. Es mejor decir que es un conjunto de protocolos que incluye TCP, IP, UDP y otros. Intentemos de todos modos dar una idea de lo que es el TCP/IP.

Para transmitir datos entre dos ordenadores tiene que ser posible identificar unívocamente a ambos; esto es posible mediante una dirección, la dirección IP, que obviamente tiene que ser única para cada uno. Veamos superficialmente el mecanismo de funcionamiento de TCP/IP. En realidad son dos protocolos distintos, TCP e IP. El protocolo IP (Internet Protocol) es el que permite la transmisión de datos entre dos ordenadores identificados unívocamente mediante su dirección IP. La transmisión de datos mediante el protocolo IP sigue un esquema muy simple. Los datos a transmitir se subdividen en paquetes de un cierto tamaño; a cada uno se le asocia la dirección del remitente y la del destinatario, y se envía el paquete.

El protocolo IP no prevé ningún control de los datos transmitidos, no comprueba que lo que se ha enviado llegue al destino ni verifica que los paquetes lleguen en el orden correcto en el que se han enviado. De hecho, puede suceder que un paquete transmitido más tarde llegue antes que otro al destino, porque la red IP ha tomado un camino más corto.

El protocolo TCP trabaja junto con IP y se ocupa de que las transmisiones se efectúen de manera correcta, comprueba que todo lo que se ha enviado efectivamente haya llegado al destino y, si hace falta, pide la retransmisión de los datos que se han perdido. También comprueba que la secuencia de recepción sea la misma que la de la transmisión. En caso contrario, cuenta con un mecanismo que ordena de nuevo la información del modo correcto.

TCP se basa en el protocolo IP para el envío físico de los datos.

>> Protocolo NetBios/NetBeui

NetBios (NetWork Basic Input/OutPut System) es esencialmente un conjunto de reglas que dictan de qué modo las aplicaciones deben acceder a la red. Desarrollado conjuntamente por IBM y por Microsoft en los años 80, se implementa de origen en los sistemas operativos de Microsoft y su funcionamiento es relativamente simple. Cada ordenador se identifica en la red con un nombre que se le da en el momento de la instalación del sistema operativo válido como dirección del destinatario. Cuando llegan datos en un paquete NetBIOS, cada ordenador conectado a la misma red recibe una copia que se descarta si no es la dirección del destinatario.

Este sistema, aun siendo simple, degrada las prestaciones de la red puesto que el tráfico gestionado de este modo estará constituido en su mayoría por paquetes destinados a ser descartados. Por otro lado, no permite comunicar directamente varias redes entre ellas y no puede acceder al exterior. En consecuencia, no sirve para una red como Internet y se destina a pequeñas redes cerradas en las que sí puede obtener prestaciones respetables si el número de ordenadores es bajo.

Tiene otras ventajas, como la casi total ausencia de procesos de configuración, a excepción de la introducción del nombre en el momento de la instalación, y el reducidísimo espacio de ocupación de memoria que necesita el software de gestión.

>> Protocolo Ipx/Spx

IPX/SPX (Inernetwork Packet eXchange/Sequential Packet eXchange) es un producto creado por Novell para sus propias redes, que se convirtió en un estándar al inicio de los años 90. En realidad se trata de dos protocolos distintos muy semejantes que trabajan juntos formando un único protocolo.

El protocolo IPX se puede considerar similar al NetBIOS/NetBEUI pues de hecho también éste prepara paquetes de datos y simplemente los envía a la red, sin preocuparse de que el destinatario esté conectado, de si los recibe íntegramente o no y ni siquiera de si efectivamente los ha recibido. Por este motivo se le




ha unido el SPX que es capaz de obtener esta información e identificar un determinado ordenador en la red a través de su nombre y una determinada dirección memorizada en el hardware de red en el momento del montaje.

Se trata de un protocolo de red robusto y sólido que sólo tiene una desventaja: no posee el control central de los nombres de los ordenadores conectados y por lo tanto puede provocar conflictos de red, aunque son raros.

Ofrece la posibilidad de hacer que varios segmentos de red se entiendan entre ellos a través de enrutadores capaces de gestionar este protocolo y de comunicar con redes Novell muy extendidas en ambientes de gestión.

>> Protocolo AppleTalk

Este protocolo fue estudiado por Apple para poner en funcionamiento redes Macintosh. La asignación de una dirección AppleTalk a un determinado nodo se realiza dinámicamente. En el momento en que se pone en marcha el ordenador el sistema elige la dirección autónomamente y envía a la red una petición de confirmación.

Si no hay ningún otro ordenador que responda a la dirección elegida se asigna autónomamente para su conexión. De lo contrario, si la dirección elegida ya ha sido asignada el ordenador recibe una respuesta de error por parte de quien ya la tiene. A continuación, el ordenador prueba con otra dirección hasta encontrar una libre. Todo ocurre sin la intervención del administrador de red o el gestor del ordenador.

Cada dirección de red AppleTalk se asocia al hardware de red del ordenador al que ha sido asignado. En el momento del envío de datos, el ordenador controla si en la propia base de datos temporal ya existe una asociación dirección-hardware similar a la recibida y si fuese así reconoce el ordenador en cuestión y acelera la operación.

Más allá de toda esta facilidad de uso, existen límites en el protocolo AppleTalk. De hecho no ha tenido mucha difusión fuera de las redes Macintosh. De entrada, el curioso File System adoptado por Apple dificulta el intercambio de archivos desde un sistema Apple a un sistema no Apple.

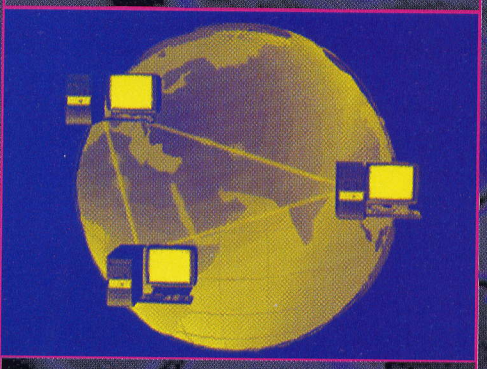
En segundo lugar, el hecho que Apple Talk fuese un protocolo propietario de Apple ha alejado a los programadores de sus secretos prácticamente cerrando el acceso por parte de tecnologías externas. Hoy las especificaciones técnicas se pueden consultar libremente con el objetivo de atraer a los programadores...

>> La red TokenRing

La red Token-Ring nace de estudios efectuados por IBM y resulta ser la red informática más rápida disponible actualmente.

Su nombre deriva de la curiosa estructura de las conexiones, en forma de anillo (en inglés ring) y de la técnica de transmisión de mensajes. En un anillo de conexión viajan varios paquetes llamados token que indican un determinado estado: libre o ocupado.

Cuando un ordenador conectado a la red tiene que transmitir datos espera hasta que recibe a la entrada un token libre, que en la práctica significa que no transporta datos. Una vez detectado este token lo substituye con un token ocupado seguido del paquete de datos que se quieren entregar y así se coloca en el círculo del flujo de red. Si el paquete de datos, una vez ha hecho todo el recorrido, vuelve al ordenador que lo ha enviado éste lo elimina del flujo y vuelve a poner en su lugar un token



libre y espera una vez más su turno para transmitir hasta encontrar el próximo token libre.

El resultado obvio es que en una red Token-Ring el transporte de paquetes de datos hasta el destinatario está siempre garantizado puesto que estos se van retransmitiendo hasta conseguir llegar, momento en el cual se eliminan de la circulación de la red del ordenador del destinatario, que los substituye a su vez por tokens libres.

Este sistema junto con la calidad de los paquetes utilizados (si pensamos en la fibra óptica) hacen de las Token-Ring unas redes de altas prestaciones.

LAS TIPOLOGÍAS DE RED

LAS REDES DE MALLA
Son redes en las cuales cada nodo está conectado con muchos otros y hasta con todos. En una red distribuida, los mensajes se mandan de un nodo a otro escogiendo uno de los muchos recorridos disponibles. La elección del recorrido se puede realizar de una manera dinámica, según las condiciones de tráfico de la red.
En cualquier caso, el recorrido de un mensaje ocupa un solo subconjunto de nodos disponibles y cada uno por un tiempo limitado.

LA RED DE ESTRELLA
Se basa en un nodo central llamado hub al cual se conectan todos los demás nodos periféricos. La comunicación entre dos nodos está centralizada y pasa siempre por el nodo central.

LA RED EN BUS
En las redes en bus todos los nodos están conectados a un cable lineal (bus), como a los afluentes de un río, mediante ramificaciones a las que se conectan los ordenadores.
En este tipo de red todos los nodos comparten un mismo canal de transmisión, y cada mensaje viaja siempre en todas direcciones.

RED EN ANILLO
La red en anillo está formada por una serie de nodos interconectados que forman un anillo cerrado.



Ethernet: Es la tecnología más extendida para montar redes locales. La desarrolló un joven investigador, Bob Metcalfe, a quien le encargaron encontrar una manera de conectar entre sí las estaciones de trabajo ALTO, los primeros ordenadores basados en iconos y ventanas (otra genial invención nacida en el PARC, difundida al cabo de diez años con la comercialización del primer Macintosh por parte de Apple).

TODO SOBRE LOS VIRUS APPENDING

Conocer los virus

¿Quieres defenderte de un virus? ¿Quizás de un appendig de los archivos .com? Es hora de conocer al "enemigo". Y para conocerlo, hay que saberlo "construir"...

No se trata aquí de infectar toda la red soltando virus como bombas sobre Bagdad. De lo que se trata es de comprender qué son los virus, cómo se construyen y cómo se procede, para poder defenderse apropiadamente ante un ataque. De hecho, conocer la técnica de construcción es esencial para determinar las contramedidas y la defensa adecuadas. Aquí trataremos sólo un tipo en particular de virus, el appending a los archivos COM. Para quienes no lo recuerden, aquí teneis un breve resumen del proceso de infección directa de un archivo COM:

Buscar un archivo de tipo COM utilizando la función 4Eh de la INT 21h (llamada FINDFIRST).

Abrir el archivo en modo lectura (función 3Dh, también del DOS)

Buscar un marcador o algún tipo de indicio para marcar un archivo como ya infectado.

Si el archivo no está infectado, seguir con la infección; si no, llamar a la función FINDNEXT (AH=4Fh del DOS) y volver al punto 2.

Estos pocos pasos son la parte inicial de un virus. No hacen nada especial, sólo buscar una víctima y "abrirla".

Este tipo de virus tiene la finalidad de adherirse al final del archivo, escribir una instrucción de salto (JMP) al principio del programa para pasar la ejecución al código del virus, y al acabar ejecutar el código del archivo original.

Para hacer todo esto es indispensable escribir el código del virus recordando que todas las referencias a variables, a punteros de tipo FAR o a otras instrucciones sin un desplazamiento relativo, se harán añadiendo a las posiciones en memoria de la instrucción o variable las variaciones debidas



a las distintas ubicaciones que ocupa el virus en el archivo.

Recordareis que dejamos la infección a medias y nos detuvimos a hablar del DTA. Decíamos que servirá para recuperar el nombre del archivo a infectar. Esto es cierto en parte, ya que en el DTA encontraremos muchísima información sobre el archivo "víctima", parte de la cual resultará útil. Una de las primeras cosas que hay que hacer cuando se escribe un virus es no usar el DTA original, sino usar una copia en memoria.

1) Identificar el archivo .com

Tras localizar un archivo COM sin infectar, nos tomaremos la molestia de dedicar algunas líneas de código para guardar datos útiles del DTA, como los atributos del archivo, la fecha y la hora de su creación.

Guardar estos datos es indispensable (**o al menos lo era antes de Windows**) para evitar que se descubra enseguida la infección. En efecto, una vez concluida ésta, los datos deberán volver a escribirse adrede en el archivo. Otro punto importante es desactivar los atributos del archivo (tras haberlos guardado): si por ejemplo el archivo fuera Read Only, el virus no se podría escribir en su interior y no habría infección. Por ello es preciso cambiar los atributos del archivo y dejarlos otra vez igual al acabar.

Una vez encontrado un archivo a infectar, empezaremos por hacerlo modificable con la función 43h/01h de la habitual INT 21h que espera en DS:DX el nombre del archivo (pero por comodidad podremos usar un LEA DX,nombre archivo) y en CX el atributo a dar al archivo pondremos 0. Dado que en Windows los virus de acción directa sobre los archivos COM son prácticamente inútiles tendremos que imaginar que trabajamos bajo DOS. En este sistema operativo algunas sutilezas, como la fecha y hora, eran muy útiles para detectar una infección pero en Windows lo son relativamente. Por ello, por una cuestión de principios, insistiremos en estos detalles.

2) Escribir el archivo

Pasemos a la infección y veamos cómo escribir en el archivo. Antes de sustituir los bytes iniciales con la instrucción de JMP al virus, hay que guardar los bytes que se sobreescribirán. El archivo debe abrirse con la instrucción 3Dh del DOS (**sintaxis en el recuadro**), y luego se procede a escribir el código hexadecimal de la instrucción JMP, que es E9, seguido por la posición del virus en el programa infectado. La dirección de memoria de nuestro virus no será siempre la misma, ya que tendremos que colocarlo en base al tamaño del archivo víctima.

La función para escribir en un archivo es la 3Eh (ver la sintaxis de las funciones en el cuadro resumen al final del artículo). La función devuelve en AX un handle (que a veces se traduce por "manejador") que servirá para identificar el archivo en el futuro. Pero como habréis observado, en la función de escritura, el handle va en BX. Por ello es más sencillo usar la instrucción XCHG AX,BX después de abrir el archivo, en lugar de MOV BX,AX que habría hecho el virus, aunque poco, más largo. Otra instrucción que se usa en lugar de MOV cuando se necesita una referencia, por ejemplo, a DI:DX, es LEA (**Load Effective Address**).

Recordad siempre que lo que distingue a un buen escritor de virus de los mediocres es saber optimizar al máximo el propio có-




digo. Es oportuno tener presente estas pequeñas abreviaturas que en los virus más "interesantes" marcan la diferencia.

3) Insertar el JMP

Una vez visto cómo escribir en el archivo, veamos cómo escribir nuestro JMP al principio del programa. Se empieza por colocar el puntero dentro del archivo al principio mediante la función 42h. Esta función pide en AL el tipo de posicionamiento requerido (ver tabla al final del artículo) en BX el handle del archivo abierto, en CX:DX la posición a ocupar en el archivo y en DX:AX se devuelve la posición donde se encontrará el puntero, al volver de la llamada. Observa que esto puede usarse como un método alternativo para leer el tamaño del archivo: si colocamos el puntero más allá del fin de archivo, sabremos cuánto mide leyendo DX:AX. Tras colocar el puntero sólo falta escribir el código del JMP (**en la forma 0E9h**) seguido de la posición del parásito.

```
MOV AX,4200h; para mover
el archivo
XOR CX,CX; pone a cero CX
XOR DX,DX ;pone a cero DX
INT 21h; de hecho CX:DX =
00:00 lo que apunta
al principio del archivo

MOV  AX,WORD  PTR  [BP+LONGI
TUD_ARCHIVO]; cálculo de la
posición del virus
SUB AX,3; sustrae del
tamaño del archivo
tamaño de la parte
sobreescrita por el virus
MOV WORD PTR [BP+CODIGO_JMP
+1], AX; dejar apartado
el código a escribir
MOV AH,40h
MOV CX,3
LEA DX,CODIGO_JMP
INT 21h
```

Este fragmento de código resume lo dicho hasta ahora. En BX se encuentra ya el handle del archivo obtenido en la apertura, se ha guardado en LONGITUD_ARCHIVO el tamaño de la víctima, y existe una variabile CODIGO_JMP que contiene el valor 0E9h.

4) Crear el Virus

Después de haber escrito el salto, hay que escribir el cuerpo del virus propiamente dicho. Con la teoría vista hasta ahora no debería ser difícil efectuar esta operación. Hay que desplazar el puntero del archivo del principio al fin. Una vez más, recorremos a la función 42h pero con AL puesto a 02h y de nuevo poniendo a cero CX y DX.

Tras haberse colocado al final del archivo, basta con copiar el cuerpo del virus y cerrar el archivo abierto. Para escribir volveremos a usar 40h y como datos a escribir bastará con indicar la etiqueta de inicio del propio virus, mientras que el cálculo del valor a dar a CX lo haremos restando al final del programa la dirección de inicio.

```
Parte_1:
     JMP Parte_2
     Marcador DB "M"
Parte_2:
. . . código del virus . . .
Fin:
```

Volviendo al esquema anterior, haremos un LEA DX, [BP + Parte_2] para seleccionar los datos a escribir, mientras con un MOV CX, [Fin – Parte_2] encontraremos el número efectivo del byte a escribir.

5) Toques finales

Hecho esto, el virus habrá infectado el archivo, sólo falta cerrarlo llamando a la función 3Eh del DOS (únicamente con el handle en BX y que cuando vuelve destruirá el contenido de AX), y restituir el control al programa original.

Antes de seguir, un recordatorio: para evitar la creación de un virus solapador (es decir, que estropea el programa infectado) debemos recordar guardar el primer fragmento del programa donde escribiremos encima el JMP al virus.

Esto se puede hacer cuando se controla un eventual marcador, leyendo los primeros 4 bytes del programa. También hay que acordarse de devolverlo todo a su sitio tras haber infectado. Habrá que modificar de nuevo el atributo del archivo para devolverlo a la situación original (de nuevo con la función 4301h) y volver a copiar la fecha y hora de la última modificación.

Todo esto se hará justo después de cerrar el archivo víctima. Queda aún una cosa por hacer: antes de infectar sería oportuno, en cuanto el control pasa al virus, guardar los registros como DS y ES.

Un método podría ser hacer un push para restaurarlos antes de que termine la ejecución del virus. Podremos restituir la ejecución poniendo a cero los diversos registros de paso (AX, BX, CX, DX) y los registros de puntero DI y SI, con un sencillo XOR, para acabar finalmente con un RETF.

También se podría hacer apuntar los punteros de instrucción a la posición actual de la primera parte del antiguo programa en el virus y luego saltar con un JMP a la misma.

Dicho esto, así concluye esta segunda parte de la creación de un virus sencillo.

Con todo lo dicho hasta aquí probablemente no conseguirás hacer un virus inmediatamente y, en efecto, el objetivo buscado no era proporcionar un código para "copiar y pegar" para convertirse en un "killer" en ciernes, sino dar la posibilidad de satisfacer la curiosidad de quien quiera saber algo más sobre los virus.

Si quieres convertirte de verdad en un escritor de virus, nuestro consejo es que estudies ensamblador a fondo (así como el C), y hacer muchas pruebas sin desanimarte ante los primeros fracasos.

Si contamos con espacio en un futuro número intentaremos tratar el cifrado y el polimorfismo o bien los virus TSR que, ya obsoletos, han representado un capítulo propio en el mundo de los virus.

[MiMMuZ]

6) Resumen de funciones

INT FUNCIÓN Descripción
21h 3Eh Cierra un archivo abierto. En BX va el handle del archivo y AX se destruye al terminar la operación.
21h 40h Escribe en un archivo abierto. En BX el handle, en CX el nº de bytes y en DI: DX los datos a escribir.
21h 42h Mueve el puntero en un archivo. En AL va el método (00h=desde el principio,01h=desde la posición actual, 02h=desde el fin de archivo). En BX va el handle, en CX: DX el desplazamiento deseado, en DX: AX devuelve la nueva posición.
21h 4301h Cambia los atributos de un archivo. En DS: DX pide el nombre de archivo y en CX el atributo a darle.



Virus: saber programar un virus es útil ya sea para defenderse, o bien para resolver problemas de programación en un sentido amplio.
JMP: Es una instrucción de salto al principio del programa que pasa la ejecución al código viral, y vuelve al archivo original.



Los fundamentos de la programación

¿Quieres hacer de hacker pero entras en crisis cuando se trata de programar el vídeo? Aquí tienes un artículo muy, muy simple para comprender el abc del... lenguaje C

Empezamos en este número a ocuparnos de la programación, una materia compleja de por sí pero que intentaremos afrontar de modo que sea comprensible hasta para los no iniciados. El lenguaje C se encuentra en la base de muchos instrumentos útiles relacionados con las redes y los sistemas: tener una mínima idea es algo casi obligatorio, y el dominio de este lenguaje es una de las cosas que establecen la diferencia entre un hacker curtido y un novato.

Para presentar el lenguaje C, realizaremos un par de cortos programas. Empezamos con uno muy simple que muestra en la pantalla el texto "¡Hola, mundo!".

Programar en c es muy intuitivo porque el lenguaje tiene pocas reglas de escritura y de sintaxis. Observando cómo se ha escrito el programa lo comprenderás enseguida.

```
#include <stdio.h>
main() {
printf("\n¡Hola, mundo!\n");
return 0;
}
```

Si escribes en un editor de textos estas líneas, no sucederá nada. Para que un programa pueda ejecutarse, primero debe compilarse, es decir, traducirse de un formato comprensible para los humanos (el c), a uno que el ordenador pueda manejar. De ello se ocupa un programa llamado compilador; el más conocido de C para Linux es Gcc (Gnu C Compiler), y se instala de modo predefinido en casi todas las distribuciones. Si tienes Linux a mano, escribe:

Gcc nombrearchivo.c

Para crear el archivo binario ejecutable, que podrá iniciarse con:

./nombrearchivo

Si, por el contrario, usas Windows, tendrás que descargar e instalar un compilador. Nuestro consejo es que utilices Borland Turbo C 2.01, que ahora es gratuito. Lo puedes descargar en http://community.borland.com/museum (tras haber completado un molesto y larguísimo formulario de registro). Tras haber escrito el listado del programa en Turbo C, basta con que pulses Ctrl+F9 para compilar el archivo ejecutable.

Si lo has hecho todo bien, aparecerá en la pantalla el texto "¡Hola, mundo!", y volverá el símbolo del sistema.

```
c:/>Hola.exe
"¡Hola, mundo!"
c:>
```

El resultado de la ejecución del primer listado.

Veamos el listado anterior línea por línea, para tener una idea sobre cómo razona el C.

Se empieza con el texto "**#include <stdio.h>**" este comando dice al compilador que incluya en el código que escribimos el archivo stdio.h, abreviatura de "standard input-output" (la h viene de header, por ahora créetelo y sigue leyendo). Gracias a este archivo tendremos a mano las funciones básicas del lenguaje de programación, sin las que el resto del programa no tendría sentido. En la segunda línea se encuentra la palabra "main" seguida de un paréntesis en blanco. Esto es una función. En particular, main es el cuerpo del programa, y constituye la función básica que debe estar siempre presente en todos los programas. Desde main se podrán llamar otras funciones, creadas por nosotros.

Justo después de main encontramos una llave, en este caso abierta. Este símbolo indica el principio de un bloque de instrucciones (funciones, bloques de decisión, etcétera). Como se adivina, la llave cerrada indica el final del bloque de instrucciones. Las encontraremos en gran cantidad, porque, junto con las tabulaciones, ayudan a mantener ordenado el código.

PRINTF es una función definida en stdio.h, que permite enviar información al flujo de vídeo para su presentación en pantalla. Esta línea del programa es la que efectivamente cumple la función de mostrar en pantalla la frase "¡Hola, mundo!".

\n es un carácter de control de la función printf, y se usa para obtener un retorno de carro en ese punto de la frase.

Observa que al final de línea hay un punto y coma. Esto es necesario en todas las líneas que contienen un comando específico (funciones y ciclos no requieren esta formalidad), e indica al compilador el fin de la instrucción.

Nuestro programa se iniciará dentro de un sistema operativo que controla todas las aplicaciones iniciadas en su interior. Para evitar problemas, es preciso insertar return 0 que, como es fácil de adivinar, hace volver. ¿Volver adónde? Al prompt de la línea de comandos. Guardamos nuestro programilla en un archivo llamado hola.c y tras la compilación obtendremos el archivo ejecutable (hola.exe en Windows), listo para probarlo.



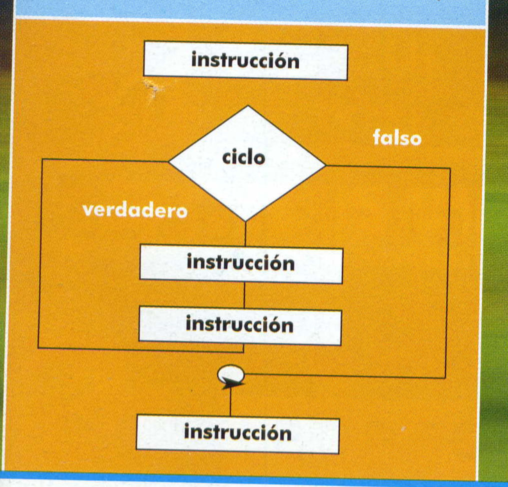
>> Las variables

Un programa que ejecuta un cálculo determinado, como por ejemplo 3 + 4, no tiene mucho sentido. Lo bueno de un ordenador, y de sus programas, es la posibilidad de ejecutar funciones y cálculos generalizados, que se adaptan a diversas situaciones. Por ejemplo, un programa muy simple puede ejecutar la suma de dos números, sean cuales sean. Para hacerlo, es necesario un método para definir la función (suma dos números) que represente el valor de los dos números de forma simbólica, generalizable. Para ello están las variables, una especie de contenedores que pueden representar números, cadenas de texto y otros datos. Si llamo A al primer número, y B al segundo, puedo crear un programa cuya función principal será algo como "A+B". Si en un punto del programa preveo la posibilidad de asignar a A y B valores cualesquiera, obtengo una calculadora que ejecuta sumas. En el ejemplo siguiente, crearemos una variable x de tipo "int" (entero), o sea, una variable adaptada para contener un número entero entre 0 y 255 (con un total de 256 valores, contando el cero), y le asignaremos el valor 0 (cero). Las instrucciones serán:

```
int x;
x=0;
```

>> Los ciclos

La disquisición sobre las variables sirve para que podamos hacer que nuestro pequeño programa escriba la misma frase diez veces. En este caso utilizaremos un ciclo, es decir, una secuencia de código que se repite mientras determinada condición sea verdadera. Cuando la condición ya no sea verdadera, el flujo del programa "sale del ciclo" y ejecuta las instrucciones siguientes. Veamos cómo usar el ciclo en nuestro programa:




```
#include <stdio.h>
main(){
        int x;
        x=0;
        while (x<10) {
        printf("\nHola mundo
        n° : %d", x);
        x++;
}
return 0;
}
```

Este listado escribe la frase "Hola mundo" diez veces, indicando cada vez un número creciente. Veamos paso a paso el código.

```
c:/>Ciclo.exe
Hola mundo n°: 0
Hola mundo n°: 1
Hola mundo n°: 2
Hola mundo n°: 3
Hola mundo n°: 4
Hola mundo n°: 5
Hola mundo n°: 6
Hola mundo n°: 7
Hola mundo n°: 8
Hola mundo n°: 9

c:\>
```

El texto producido por la ejecución del segundo listado.

El programa es idéntico al anterior hasta la palabra "int", que ya hemos visto que sirve para definir una variable x de tipo entero entre 0 y 255. A continuación, se asigna a x el valor cero. While es una palabra clave que indica un ciclo (significa "mientras" en inglés). Mientras la condición indicada dentro del paréntesis sea verdadera, el ciclo se repetirá ejecutando las instrucciones indicadas entre las llaves. En este caso, la condició se representa por x<10. Mientras x es menor que 10, se ejecutarán las intrucciones vistas anteriormente en la pantalla. Observando atentamente, se observa que el printf utilizado aquí es distinto del caso anterior. Encontramos la expresión de control "%d", que indica en qué punto deberá insertarse el valor de una variable, citada tras las comillas y precedida por una coma. Sigamos adelante, y pasemos a la línea siguiente. la expresión x++ toma la variable x y la incrementa en una unidad (por ello, a partir de este momento, x será igual a x+1).

Intentemos hacer un resumen de lo visto hasta ahora:

*** se define una variable x de tipo entero e igual a cero.**

*** mientras esta variable es menor que 10, el programa muestra en pantalla la frase "Hola mundo" y al lado escribe el valor actual de x.**

*** el valor de x se incrementa en 1.**

*** el ciclo se encierra entre llaves, y por ello vuelve al principio.**

*** tras 10 ciclos, x será igual a 10, y la condición indicada en la expresión while ya no será verdadera. El ciclo termina, pues, y la ejecución prosigue con las instrucciones sucesivas.**

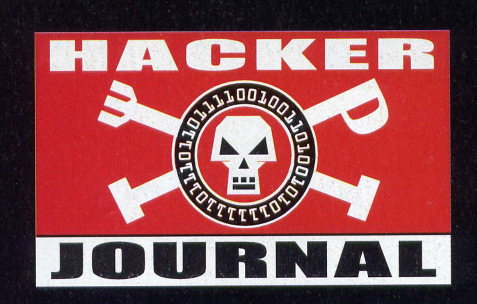
Como antes, guardamos el archivo con el nombre ciclo.c y compilamos obteniendo el ejecutable. Ya con el programa, observaremos que el valor de la variable, lógicamente para el ordenador, para nosotros es algo desconcertante: nosotros contamos de 1 a 10, y no de 0 a 9. >Para resolver este problema, basta con inicializar la variabile x a 1 en lugar de 0, y poner como condición x<11.

INCREMENTAR Y VERIFICAR LAS CONDICIONES

Hemos visto que la expresión x++ sirve para incrementar x en una unidad. Del mismo modo, x-- decrementa x en uno. Además, en el ejemplo ciclo.c hemos utilizado como condición x<10. Obviamente esta es una de las muchas condiciones que se pueden verificar. Veamos las más importantes:

<	menor que
>	mayor que
<=	menor o igual que
>=	mayor o igual que
==	igual a
!=	diferente de



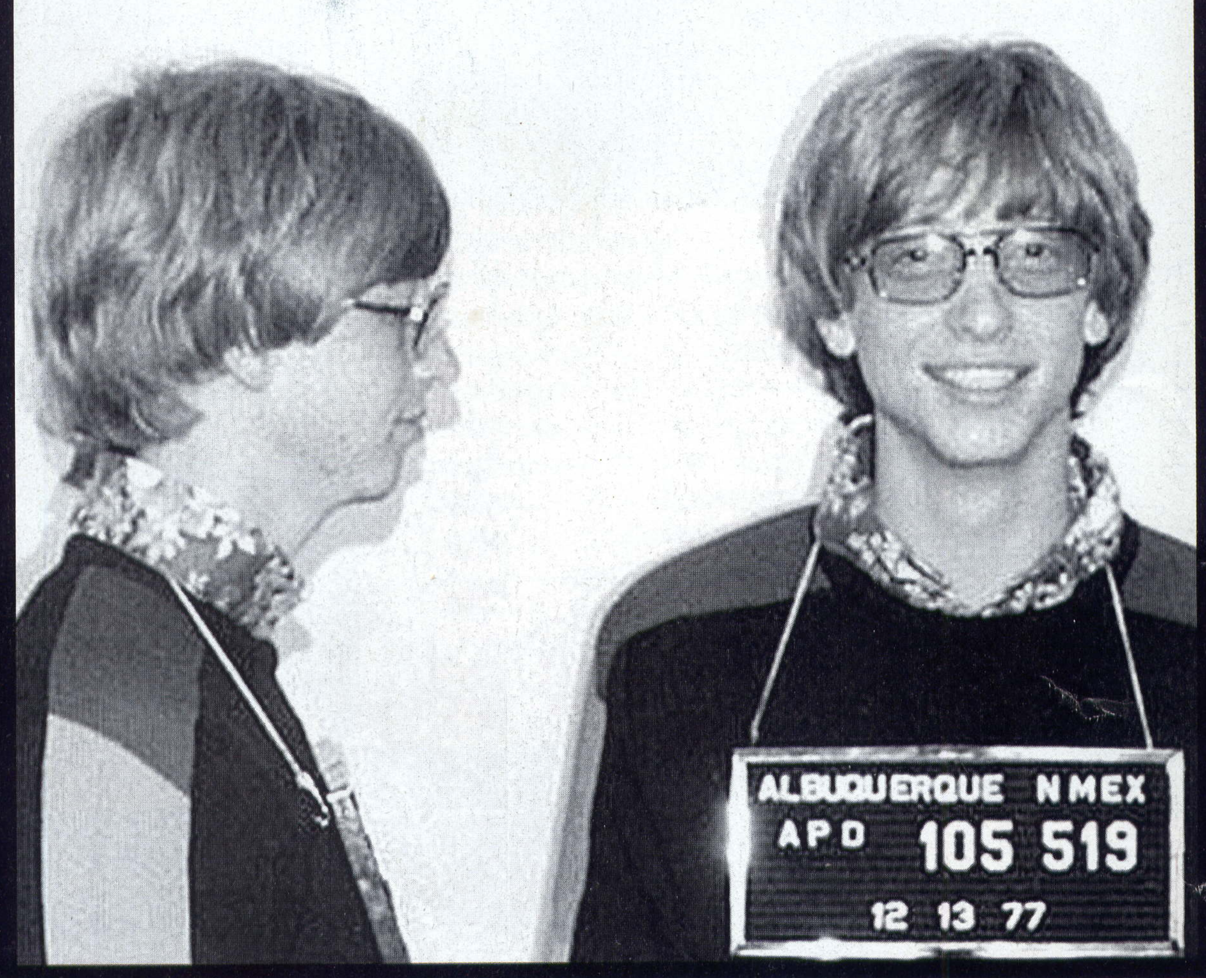






U.S. Department of Justice
United States Marshals Service

WANTED
BY U.S. MARSHALS



EL PRÓXIMO NÚMERO EN EL KIOSCO

EN DOS MESES

¡RESISTID SIN NOSOTROS!